The Thesis Committee for Robert Blake Anderson
Certifies that this is the approved version of the following thesis:

Development of Mobile Platform for Inventory and Inspection Applications in Nuclear Environments

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: ______________________________
Sheldon Landsberger

Co-Supervisor: ______________________________
Mitch Pryor

Development of Mobile Platform for Nuclear Environments at Los Alamos National Laboratory

by

Robert Blake Anderson, B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2015

Acknowledgements

I am grateful to the following people without whom I would not have completed this program:

- The Los Alamos MET-2 group for their financial support of myself of the UT Nuclear Robotics Program.
- My academic advisors Dr. Mitch Pryor and Dr. Sheldon Landsberger for their guidance the past three years.
- The other Nuclear Robotics Group members whose work has contributed to the LANL vault project.
- The University of Texas nuclear faculty, in particular Dr. Erich Schneider and Dr. Steven Biegalski.
- My former engineering co-workers at Nuclear Logistics, Inc.

Abstract

Development of Mobile Platform for Nuclear Environments at Los Alamos National Laboratory

Robert Blake Anderson, M.S.E

The University of Texas at Austin, 2015

Supervisor: Sheldon Landsberger

Co-Supervisor: Mitch Pryor

The efforts made towards deploying a mobile robotic system at Los Alamos National Laboratory are detailed in this thesis. The platform application is non-contact tasks related to inspection, inventory, and radiation surveying. It is intended for a Special Nuclear Material storage facility featuring a high radiation environment and a variety of storage modes.

New robotic capabilities have been developed using several mobile platforms to address the requirements of this application. Many of challenges are common to any warehouse application, such as autonomous task planning, vision, navigation, and inventory data management. Others are specific to a nuclear laboratory environment, such as radiation measurement and analysis, response to radioactive contamination, criticality safety, and restrictive security measures. This thesis describes the progress made towards meeting these challenges, outstanding issues, and future work that is necessary to complete the project.

Nuclear facilities are under ever-increasing demands to reduce worker radiation exposure. Since the *vault* is a high radiation area, it is one of the first targets at Los Alamos for the application of novel solutions. The deployment of this system promises to enhance worker safety by reducing their presence inside the vault and therefore total occupational dose. As robotic systems become more trusted in the nuclear weapons complex, it also has the potential to reduce total operator labor by performing time-consuming tasks autonomously.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Hazards of Nuclear Work and Potential for Robotic Mobile Platforms

Industrial nuclear processes often require human presence in radioactive or toxic environments. The nuclear weapons complex in particular faces significant worker safety challenges due to the highly radiotoxic and chemically unstable nature of plutonium and other actinides. Nuclear workers are subject to stringent annual limits on occupational radiation dosage, currently 5 rem [0.05 Sv] per year for commercial nuclear workers at Los Alamos National Laboratory (LANL) [1]. Workers are required to wear personal radiation monitoring equipment, and dosage is closely tracked to insure that regulatory limits are not exceeded. Mobile robotic systems are a topic of growing interest in the nuclear power and research fields. Machines are generally far more resilient than organisms against radiation damage and do not carry the same ethical and legal considerations that apply to human workers. Therefore, facility operators can solve the challenges related to hazardous environments by replacing human labor with robotic labor.

Ever-increasing regulatory requirements for worker protection stretch the ability of facility operators to fulfill mission requirements using traditional principles. The traditional goal of exposure reduction is “As Low As Reasonably Achievable (ALARA), which is implemented by the following methods:

- Reduce the time that personnel are in the radiation field. Total dosage is proportional to the exposure time.
- Increase the distance between the personnel and the radiation source. Dose rate from a point source decreases according to the inverse square ($1/r^2$) of the distance.
- Apply shielding material between the personnel and the radiation source.

Facility operators have spent a great deal of effort applying the traditional methods, but because of diminishing returns it becomes ever more difficult to achieve dosage reductions without compromising the facility mission. A need for novel solutions to limit dosage drives research into robotic technologies. Robotic systems have the potential to reduce or eliminate the need for human workers in the hazardous areas of facilities. This achieves dosage reduction via the ALARA method. Robotics has long been a subject of interest to the Department of Energy (DOE) [2], but the past ten years has seen greatly increased capabilities of robotic systems and increased utilization in the field. Possible applications for mobile systems include nuclear material transport, visual inspection of material and equipment, security safeguards, background radiation and contamination surveys, and responding to contamination events.

1.2 Los Alamos Vault Configuration and Environment

Los Alamos National Laboratory (LANL) has been the Department of Energy's primary weapons research and production facility since the end of the Cold War. With the closure of Rocky Flats Plant in 1992, LANL was tasked with the mission of small scale primary-stage pit manufacturing in addition to its traditional design and testing activities. Production supports the stockpile stewardship program and helps maintain a technical knowledge base should mass production of nuclear weapons become necessary again [3].

Plutonium chemistry, machining, and welding operations take place in the plutonium production facility (PF-4) at LANL. In PF-4's basement there is a secure vault for storage of nuclear material. The vault contents include legacy material dating back decades as well as material used in current processes. The bulk of the material is plutonium in various concentrations, chemical compositions and ages. Highly Enriched Uranium (HEU) and ^{252}Cf make up smaller portions of the vault inventory. One of the vault chambers is dedicated to storing ^{238}Pu for radio-thermal generator applications.

The vault is one of the most intense radiation environments at LANL. Personnel in the areas with the highest measured dose rates will reach their annual dose limit in a matter of hours. The radiation field is dominated by neutrons from spontaneous fission, with gamma rays from fission and daughter products as a minor component. Beta and alpha radiation does not penetrate the material containers. Surface contamination threats are primarily from plutonium (^{239}Pu) or uranium (^{235}U and ^{238}U) oxides and from americium (^{241}Am), all of which are alpha emitters. Therefore contamination surveying is based on alpha detection.

Material is deposited in – or retrieved from – the vault multiple times during a typical work day. Transport operations are performed by a worker in "anti-contamination" clothing. Personnel movements in the vault are planned in advance to reduce the operation's duration and avoid the highest radiation areas. Containers undergo an assay procedure after being removed from the vault with radiation readings taken at 30 cm. Containers range in size from 5 kg canisters to 55 gallon drums. Decades of operations have resulted in a wide variety of container types and labeling standards in the vault, however LANL is in the process of standardizing the containers to a single series of models and labeling syntax [4].

Containers are stored in several types of cabinets and drawers, with illustrative nicknames such as:

- "Filing cabinet" type drawer in which containers are upright.
- "Wine rack" style, in which containers lie in angled recesses.
- "Cage shelves", in which containers are placed side by side on shallow shelves with a meshed, hinged cover.
- "Bath" receptacles in which containers are placed in metal mesh baskets and lowered into circulated coolant. This is used for material that produces large amounts of decay heat, namely ^{238}Pu.

Some containers are simply placed upright in designated floor sections. This is mainly used for the largest containers that would not fit in cabinets. Hinged gates secured by pins prevent the containers from rolling out into the aisle in the event of an earthquake.

The vault layout consists of a main passageway with storage chambers branching off. The floor is cement with a very small incline for drainage purposes. The chamber doors are very heavy, too heavy for any existing mobile robot to open. The chambers contain the various storage means listed above. The vault is climate controlled at room temperature and brightly illuminated at all times.

1.3 Robotic Development Platforms

Robotic capabilities have rapidly advanced in recent years as the power, miniaturization, and cost of on-board computer and vision systems have improved. The Nuclear Robotics Group (NRG) at The University of Texas at Austin has procured several mobile platforms of increasing sophistication (Figure 1-1). These platforms were used for development of algorithms and technologies applicable to mobile tasks in laboratory or industrial environments.

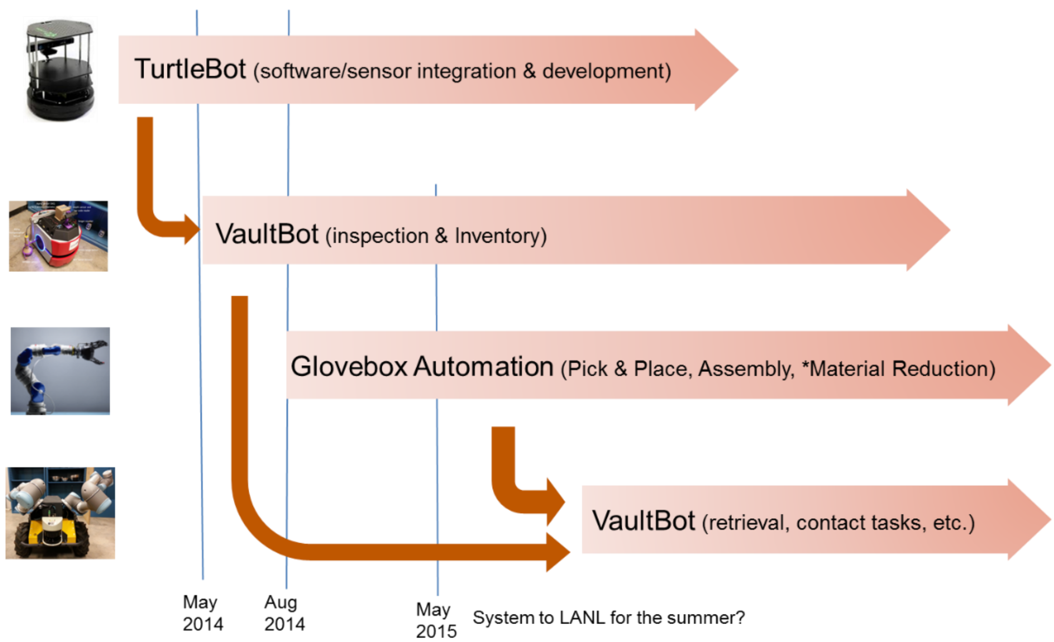


Figure 1-1: NRG Development Roadmap

1.3.1 Clearpath Turtlebot 2

The first platform acquired by NRG was the Clearpath Turtlebot 2 [5], a small, two wheeled, differentially steered mobile system (Figure 1-2). It is equipped with three front bumpers, a cliff sensor, and wheel drop sensors which help prevent it from knocking over objects or falling off ledges. The Turtlebot also features RGB and depth vision provided by a Microsoft Kinect. The Kinect represents a major improvement in 3D visual resolution and range over the systems that had previously been used by roboticists, and this allowed for more reliable and faster autonomous navigation. The Turtlebot also features three different flat mounting surfaces for accessories. Thus, although the Turtlebot provides little payload capacity and no manipulation, it was useful for early experimentation with autonomous navigation and sensor integration. Early radiation surveying demonstrations were also developed using this platform as detailed in Section 5.6.

Figure 1-2: Clearpath Turtlebot

1.3.2 NRG Vaultbot

The second system is known as the NRG Vaultbot. It is built on a Clearpath Husky base [6], which is a high payload, four wheeled, skid-steered platform. The Husky is far more robust and powerful than the original Turtlebot, thus addressing the main restrictions faced in early development.

Figure 1-3: NRG Vaultbot

A pair of Universal Robots UR5 manipulators [7] are mounted on top of the Husky (Figure 1-3). The UR5 is a 6-Degree of Freedom (DOF) arm with a 5 kg payload. This is sufficient for mounting grippers and handling small objects. The arms are powered from the Husky battery and feature a teaching pendant also mounted on the platform. The teach pendant is used for startup and calibration operations, and can be used by a human to teach movements by manually moving the joints. A SICK LIDAR system is equipped on the front of the base for 3D vision.

For manipulation tasks, the UR5s are fitted with Robotiq 3-Finger Grippers (Figure 1-4). The gripper weight of 2.3 kg takes up just under half of the arm payload. They are powered via integrated power supply connections at the ends of the arms. Vision systems have also been mounted to the end-effectors for use in inspection tasks, which will be documented in a forthcoming publication.

The main limitations of this platform are the short battery life and difficulty with autonomous navigation. When the drive and manipulators are in heavy use the 20 Ah battery is quickly exhausted, typically in about half an hour. The skid-steered drive does not work well with the autonomous navigation algorithm since it relies on accurate odometry to track the robot position. In skid-steering, the odometry does not correspond to the movement when the robot is turning.

Figure 1-4: Robotiq 3-Finger Gripper [8]

1.3.3 Adept Pioneer LX

Non-contact development has recently moved to a new platform, the Adept Pioneer LX [9]. The Pioneer features greater payload and a more extensive sensor suite, including an Asus depth sensor, RGB camera on a pan-tilt unit and ultrasonic forward and rear sensors. The greater payload capacity allows for testing with additional accessories such as larger radiation counters and an elevation system called the ZipperMast [10]. This system was heavily customized for the proposed the proposed task and these are outlined in Section 5.1.

Figure 1-5: Adept Pioneer LX with NRG Accessories

1.4 Objectives

This thesis presents development work which advances LANL's ability (both technically and institutionally) to complete non-contact tasks in a nuclear material vault in a plutonium production facility. Some of the challenges that must be met for the initial deployment of the system are safe navigation (either user controlled or autonomous), radiation sensing, object recognition, security, safety, and database interface. Multiple robots are utilized as a part of this effort. Contact tasks such as deposition and retrieval of material canisters into and out of the vault is an ongoing but separate effort by the Nuclear Robotics Group at UT Austin.

The initial system should be capable of a range of autonomy in terms of its maneuverability, from being teleoperated by workers in the control room outside the vault to full autonomy. It includes sensors and advanced algorithms to improve its autonomy as well as provide additional safety mechanisms for a user-operated system.

Once deployed, the system must positively address worker safety by reducing the time spent in the vault without a burdensome impact on the duration, cost or reliability in completing the necessary tasks. The vault contains very intense radiation fields, and personnel assigned to it can receive unnecessarily high doses. Total annual dose from the vault, recorded from personal radiation detectors, is typically the equivalent of several worker-years of allowable dose. Tasks must be performed quickly and personnel are rotated to avoid exceeding exposure limits. The use of robotic systems will significantly reduce the need for such onerous practices and provide greater flexibility to radiation workers to participate in other activities.

Finally, increasingly sophisticated automation will reduce labor requirements. Both of these tasks consume many worker-hours and result in significant doses. Other tasks include the radiation and security inspection performed at the start of each work day to clear the room for operations. Machine automation also promises greater precision and repeatability that a human worker can achieve.

Specifically, this report will focus on the following areas:

- Chapter 2 will discuss the requirements that the vault system must fulfill in order to qualify for live deployment in a hot environment. This includes both technical requirements as well as bureaucratic, regulatory, and procedural considerations. Since a key goal is ensuring the system is accepted at LANL for long-term use, the attention to documenting the requirements should be comprehensive and – whenever possible – quantitative.
- Chapter 3 provides an overview of previous work in the areas listed above as well as any other relevant efforts in robotic inventory or security applications.
- Chapters 4 presents the algorithms and software modules that were written to implement the system functionality needed required for the vault application.
- Chapter 5 presents the developed integrated demonstration testbed including its hardware, sensing modalities, and a brief review of other necessary technologies including supervisory control, vision and localization/navigation.
- Chapter 6 presents detailed results of inventory demonstrations using various levels of autonomy ranging from teleoperation to higher levels of autonomy. Demonstration developed at both UT and LANL will be included to accommodate the differing requirements (i.e. no wireless communication and possible no cameras at LANL)
- Chapter 7 will summarize the work and provide and outline of the technical and compliance issues that remain to be addressed as well as suggestions on how to address them and concluding with the an analysis on the feasibility of the system as a whole.

In parallel to the technical solution development, a concerted effort will be made to address the technical challenges in such a way that the institutional concerns are also addressed. Unavoidable technical and institutional conflicts will be minimized, but documented as they occur. A summary of such conflicts will be included in the summary chapter along with suggestions for resolving such conflicts.

Chapter 2: Major Challenges

2.1 Summary of Major Challenges

The barriers to deployment of robotics in nuclear environments consist of institutional factors as well as technical problems. Research efforts like this one tend to focus on only the technical issues and ignore institutional factors. Because of this, a lot of promising and useful technology is never used. Members of the Nuclear Robotics Group (NRG) typically spend their summers at LANL, and then also work as full-time lab affiliates during the final stages of the doctorate. Students historically have had sufficient access to properly document and understand the technical (nuclear and automation) and institutional issues. Thus, this effort will attempt to address the issues in both major categories to ensure a system can be realistically deployed. This chapter outlines the outstanding requirements of the application and the human environment of LANL that the system must fulfill.

2.2 Institutional Factors

2.2.1 Inertia/Risk Aversion

The biggest obstacle to the adoption of robotics in the nuclear world is that they must compete with long established, *tried-and-true ways* of doing things. This fact combined with the inherently hazardous and difficult nature of nuclear work causes hesitance among facility operators to seek out new technologies. Alternative approaches must demonstrate not only that they work, but that they credibly offer significant additional benefits to justify the risk of abandoning the existing, known solution.

This problem can be exacerbated by the “stove pipe” delegation of roles and responsibilities at a large and complex institution such as LANL. Different areas are responsible for a variety of risks from ergonomic injury, criticality, fires, worker safety, security, structural integrity, etc. Thus a variety of people with diverse backgrounds and interested must be educated with respect to technology, and accept that the potential value of the system with respect to one risk area (i.e. operator dosage) is an overall benefit even if their area (i.e. security) was carry more risk.

2.2.2 Lack of Trained Personnel

One of the implications of the fact that robotics is a novel technology is that most personnel in nuclear facilities have no prior experience using or even working near them. All tools and equipment involve training costs, but the fact that robots are uncommon and highly complex means that this is more onerous that more familiar equipment. In addition, the training methodology for robotic operators is itself an immature field.

2.2.3 Incompatibility with Existing Regulation

The current regulatory structure of the DOE complex is written with the assumption that labor is performed by humans. Establishing legal accountability via a paper trail for all work performed is one of its main purposes. The existence of autonomous robots operating without human supervision raises unforeseen questions that must be accounted for in regulatory revisions. Since facility operating procedures must be compliant with the legal regulations, this creates ambiguity and reluctance to adopt highly autonomous robots.

2.2.4 Culture of Safety and Security

In the DOE Weapons Complex, safety and security considerations are of paramount importance. These requirements take priority even over the primary mission functions of the facility. This is demonstrated by the current shutdown of Special Nuclear Material (SNM) operations at LANL, initiated in June 2013 in part due to concerns about inadequate criticality safety [11]. Robotic systems must be capable of meeting the high standards set by LANL. Just as importantly it must be able to demonstrate this capability convincingly. Robotics must overcome the fact that new technology is always subject to higher scrutiny than familiar methods, and may in fact be held to higher standards. Even a minor incident which otherwise would not cause a great deal of alarm could be fatal to the adoption of automation. Worker injuries and accidents are included in the contract between Los Alamos National Security, Inc. (LANS) and the Department of Energy as criteria for performance pay to LANS. Thus there is little tolerance at the corporate level for unproven technology which risks the bottom line.

Caution and extensive, incremental testing is necessary to prevent such an accident and to inspire confidence from the facility management. Ideally a robot will be able to perform the same training regimen that human workers are subject to, and meet the same acceptance criteria to which human workers are expected to meet.

2.3 Technical Challenges

Robot Operating System (ROS, see Chapter 4) will be used to develop the proposed system and supervisory algorithms. ROS is a set of open source software packages that allow for robotic developers to avoid “reinventing the wheel” a once common habit in academia and still often seen in the DOE complex. For this system, we will be able to utilize ROS to address the following challenges with little or no modification on our part.

- Radiation sensing
- Object identification
- Operator interface/awareness
- Communication

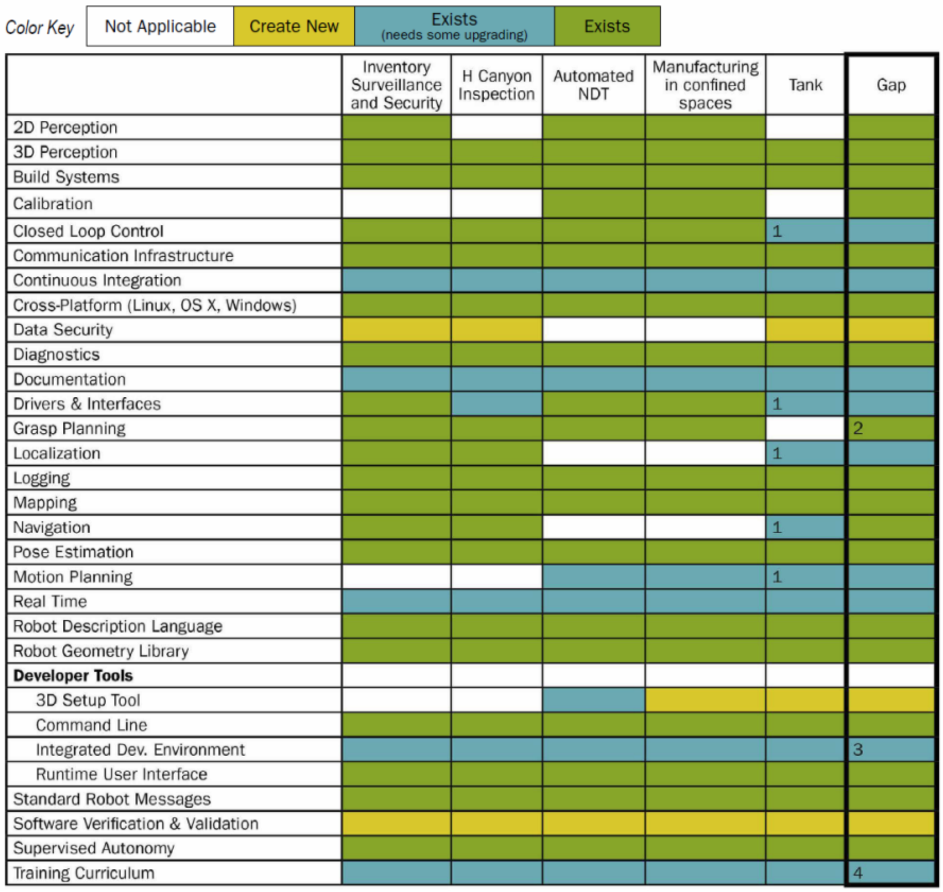


Table 2-1: Gap Analysis

This provides a starting point for determining areas where further work is needed to develop new robot capabilities. Some application-specific areas not included will need further work, however. The following sections discuss these challenges. In general, the most difficult ones are:

- Reliable object identification.
- Shared autonomy/recovery from unplanned events.
- Communication.
- Recovery localization.

2.3.1 Radiation Sensing

The system must be capable of performing radiation measurements in several contexts. It is intended to replace human effort in the quarterly ambient radiation surveys performed inside the vault. This consists of recording the neutron and gamma intensities over the entire vault. Typically measurements are made at approximately waist-height in a square grid pattern. At a minimum, the system must perform these surveys either teleoperated or autonomously. Autonomously, the robot must travel to a series of predefined map points and take a count measurement at each. Statistical uncertainty of the measurement must be accounted for by the program. Once a survey method using the mobile platform is established, it will likely be desirable to modify the survey plan which no longer must account for the dose limit of the operator.

The system must also perform radiation assay of containers in the vault. This is typically done at a distance of 30 cm to verify that the radiation emissions are consistent with the expected contents of the container.

Finally, the system must be capable of performing checks for contamination on the floor. Contamination material would most likely consist of oxides of ^{239}Pu, ^{238}Pu, ^{240}Pu, ^{235}U, ^{238}U, or ^{241}Am, which are all alpha emitters. The system must be able to account for statistical uncertainty in order to distinguish significant peaks from the background. When contamination is detected with sufficient confidence, the system must alert human operators with relevant details such as location and count rate, and the robot must avoid spreading it further.

Other issues associated with radiation sensing may include the ability to "investigate further" which is intuitive for a human operator but would require the ability to re-plan for a robot. Surveys must be planned to account for battery life and restrictions on communication. Finally, the robot must able to robustly operate in the envisioned radiation fields. While this will likely not be an issue given the relative dosage magnitudes found in the vault, analysis will be necessary to support this notion. While these capabilities need to be added to ROS, they are generally well understood in the literature [12] [13].

2.2.2 Object Identification/Inventory

The system must be capable of identifying key objects in the vault such as containers, drawers, and cabinets. Identifying containers is especially important for inventory operations, since the system must confirm that material is in the correct vault locations.

Currently, most of the LANL vault containers have paper labels affixed to the sides. Information on the labels is a mixture of handwritten and printed text. Some are simply labeled with their ID number written on a piece of tape, or with tags attached by string. Vault personnel are in the process of standardizing the labeling with entirely printed ones with a common format affixed to the side. Human workers locate containers by first looking up the cabinet number and the type of container in the vault database, going to that cabinet, and checking the identifying numbers on the labels of the containers inside. The process is entirely suitable for a human to efficiently find a given container. However, it would be quite difficult for a robot to perform this procedure. Text recognition is generally more difficult for machines, due to detrimental conditions such as insufficient resolution, blur, occlusion, poor and variable lighting, use of different fonts, and mixture of printed and handwritten text. It would also be difficult to account for the current variety of labeling methods. Alternate methods of identifying key objects are thus needed. This will most likely involve changes in the vault, such as different or additional labeling.

The principal criterion for the recognition system is accurate identification. Misidentifying a container could lead to hazardous events or cause material to be misplaced or incorrectly retrieved from the vault. Vault operators will have to investigate in person if the system incorrectly concludes that a container is in the wrong location, which will be time consuming and result in a radiation dose. The use of multiple redundant identification methods would reduce the error rate of the system.

Pose estimation of the objects is also required so that the robot can navigate to correct drawers and cabinets and locate containers for close-up inspection and eventually grasping. The robot must be aware of the container relative to itself and the vault map frame.

Once identified, the system should be able to retrieve information about the object from the vault inventory database. This will reduce workload in inventory inspections and instantly provide operators with the details needed prior to a retrieval operation. The system should be able to verify the state of the container relative to the inventory database and flag discrepancies.

2.3.3 Autonomy/Task Planning

Autonomous task execution is planned for post-deployment upgrades to the system. The vault application involves complex, multistep tasks in a high-uncertainty workspace. Therefore, a sophisticated and adaptable task specification and execution process is needed. It must be capable of taking task specifications from an operator in human language syntax and translating it into sequences of system operations. It must also be able to recognize unexpected conditions or failed operations and modify the task plan to account for them.

The challenge lies in designing an algorithm and worldspace model which is flexible enough to envelop all forseeable conditions, yet deterministic enough to generate repeatable behavior. The designer must also give thought to how the system should respond when faced with conditions that were *not* forseen, and are not explicitly accounted for. Trade-offs between safety and performance must also be modeled in the system, with the recognition that the correct decision in one situation may not be so in another situation.

2.3.4 Operator Awareness/Interface

The system must offer the operator a useful picture of the robot's situation and surroundings. The operator needs a clear and accurate idea of the robot state and configuration. When sharing control with autonomous systems, the operator needs a preview of what the system intends to do before motion begins. After performing an action, the operator needs feedback that it was performed as expected.

The most important consideration is the use of cameras for visual information. They must provide views of the immediate front of the robot and help the user judge clearance between obstacles for driving purposes. The user should be able to change the field of view of cameras using pan-tilt units or other systems. Moveable cameras should be placed such that there are minimal obstructions by other hardware on the robot base.

The platform should include sensors to detect contact with obstructions or when the robot is stuck. The output must be presented clearly enough for the user to diagnose driving problems and determine how to extract the robot from the situation.

Relevant information about key objects in the environment should be readily available to the user. The system must be capable of querying information about a container such as its location and contents and providing it in a human-readable format. The operator should also have the ability filter information to find the data that are relevant to the situation.

2.3.5 Communication

The system must be capable of communicating with a workstation outside the vault. In teleoperated mode this would consist of vision data to the workstation and commands from the user. In autonomous mode, this would consist of queries to the inventory database and alerts to the vault operators. Quality vision and depth data are require a great deal of bandwidth, typically several megabytes per frame. The other planned features are not expected to involve significant transmission loads.

Wireless communications, if allowed, must be encrypted and conform to facility requirements such as maximum allowable broadcast power and communication protocol. Wireless communications are not expected to be approved within the next few years, so it is likely that the initial deployment must use tethered communication or be required to perform tasks with full autonomy instead. Past experience with field robotics has shown that tether snags are a significant threat (see Section 3.5). Measures to mitigate that threat are needed.

Audio communication is an option for sending simple queues to human operators in the vicinity, since humans are expected to supervise it during early deployment. This could include low battery warnings, fault messages, and discovered hazards such as radioactive contamination. Text-to-speech software is installed on the Pioneer LX which could be used for this purpose. Audio communication could also be used for simple commands from an operator to the robot, such as “Go home” or “Stop.” Some simple voice commands have recently been implemented by another NRG researcher in a manipulation task context.

2.3.6 Safety

2.3.6.1 Co-Robotic Operations

The system must be capable of operating in close proximity with human workers. All industrial manipulators carry a risk of injury from crushing people against surfaces. Collision detection on the manipulators is a critical feature.

Emergency stop buttons should be installed on the robot body and at the remote workstation. They should be located in order to maximize the ability of a pinned worker to reach one of them.

Workers who are to be in close proximity to the robot while it is active should be required to complete a lab-approved safety training course. This would involve information regarding the hazards that can be posed by the robot, the safety systems which are built it, the locations of the E-Stop buttons, the meaning of any audio messages, and the appropriate response to foreseeable accidents.

The system should include contact sensors to detect collisions between the body and environment. This is important to insure that the robot will not knock over canisters on the floor or injure human workers. The Pioneer LX has bumpers and also uses its ultrasonic sensors for collision avoidance.

The platform and manipulators must be compliant with the LANL Lockout/Tagout program [14], to prevent unplanned activation of the system. A simple, compliant method for shutdown could involve removing the battery and installing a lockout device which prevents the battery from being reinserted.

2.3.6.2 Criticality Safety

Operations in the vault are subject to strict criticality safety procedures. Each inventory location has a specified material limit which cannot be exceeded. Containers are not taken into one of the storage chambers unless it is intended for deposition and has been evaluated against the criticality limit of the intended destination. As mentioned in the section above, contact sensors should be included on the body to prevent it from knocking over a container. A loose rolling container could create a criticality violation as well as risking a radioactive release.

The system should be capable of performing verification of criticality limits of individual locations during inspection. This would involve identifying the containers present, querying the inventory database for their contents, and running a check against the location's material limit.

The construction of the system must avoid using materials that have high reflectivity with respect to neutrons (graphite, beryllium, etc.), since placing such materials close to fissile material could potentially trigger a criticality event.

2.3.6.3 Recovery from Unplanned Events

The robot must be cognizant of its situation and react accordingly in all cases (it is contaminated, the battery is low, it is lost, it is performing a survey, it is doing inventory) and in different control modes (operator control, shared control, autonomy). The system needs an organized control framework for tracking the environment via sensor input so that correct behavior is selected (see Section 4.5).

The system needs to be tolerant of communication gaps, especially if wireless communication is used. The vault's thick concrete walls make signal deadzones likely. The robot needs to be programmed with autonomous behaviors that are dependent on the current task and situation. This could involve canceling the task and driving back towards the docking station until the signal is re-established. It may simply mean playing a loud audio queue and waiting for help. An autonomous recovery localization routine is necessary to ensure the robot can find its way home after a fault.

When the unplanned event has safety implications, for example radioactive contamination, the first priority is to not make the situation worse. Then it needs to alert vault operators so they can take action. Completing the robot's work tasks can wait until it is resolved.

2.3.7 Security

Security is a critical consideration since much of the material stored in the vault is classified. Thus, communications between the robot and workstation must be secure against interception. At present, no wireless communications are approved for use in secure areas at LANL. Therefore, it is likely that the early versions of the platform will need to use hardline communication or perform its task autonomously. LANL is in the process of exploring wireless communication in classified applications, so it is possible that it could be implemented later.

The sensors installed on the platform must be capable of secure use. Vision systems and microphones are especially problematic as images of the vault are considered sensitive information. The platform must include physical security features to prevent tampering or misuse. Lockouts to prevent unauthorized access to the power button and data connections are a desired feature.

The workstation must be treated as a classified machine which imposes additional requirements (i.e. strict user authentication, network connectivity, devices it can communicate with, etc.). The supervisory computer for the mobile platform cannot be connected to the internet either directly or be connected to devices that are themselves connected to the internet.

2.3.8 Summary of Requirements

The specific technical requirements for the system that are derived from these challenges are shown in Table 2-2 below.

Technical Area	Application Specifications	Safety & Security Considerations
Radiation Sensing	Can detect floor alpha contamination via non-contact sweeping. • Sweep rate as specified in LANL procedures. • Count efficiency and dead-time per LANL instrument specifications. Can measure gamma & neutron field dose rate in air. • Count efficiency and dead-time per LANL instrument specifications. • Can position instrument using mast or manipulator arm. Can create 3D radiation maps. • Can localize data by transforming instrument reference frame to world map. • Can visually display spatial radiation data for operator. • Can perform pre-defined survey plans autonomously. • Can compare radiation histories to present readings.	Does not track alpha contamination around. Check wheels/treads for alpha upon leaving vault. Decontamination plan must be in place. System behavior must be compliant with vault criticality procedures. Alarm functionality for anomalous measurements, including audible alarm for nearby personnel.
Object Identification/Inventory	Identify containers for inventory purposes. Pull information on containers from inventory database. Modify database entries due to vault activities. Misidentification rate not to exceed 0.01% when using combined identification methods per inspection procedure.	Notify operators when containers are missing or misplaced. Database connections must utilize authentication methods to prevent malicious read/write operations.
Autonomy/ Task Planning	Procedurally generate action plans for autonomous tasks. Respond to interrupt conditions and re-plan accordingly. Account for costs of actions in planning.	Planner world model includes variables to track safety conditions. Actions have dependencies on these variables.

Table 2-2 Continued

Operator Awareness/ Interface	360 degree visual field around robot using cameras capable of panning. Provide previews of autonomous actions in shared autonomy mode. Give feedback on results of actions and state of robot. Context-sensitive information display. Manual information filtering features.	Workstation is treated as classified computer with normal LANL authentication procedures. Proximity and contact sensors on base. Force compliance on manipulators or other articulating hardware.

Table 2-2: Table of Technical Requirements

Chapter 3: Literature Review

This chapter discusses previous work in various fields of robotics that are relevant to the LANL vault application. The focus is on finding existing solutions to the challenges presented in Chapter 2. Some solutions that were originally developed for other applications can also be adapted to robotics. Some complete commercial systems are presented that provide real life examples of some of the needed technologies. Finally, this chapter presents a history of mobile robotics in nuclear environments and the main difficulties/lessons learned from those experiences.

3.1 Radiation Surveying

LANL has sponsored prior work in robotic radiation surveying. Cortez *et al.* performed tests of a small mobile robot for finding floor contamination [15]. The tests focused on low-rate counting in which Poisson statistics are significant (see Section 4.2.1). The test is designed to demonstrate a search for discrete sources of radiation. The test area is broken up into square tiles, and the robot performs a sequential search of the tiles. Interestingly, the robot is never stationary while taking measurements and the robot lowers its drive speed when the count rate for a given tile exceeds a specified detection threshold. The moderated drive speed allows it to achieve lower uncertainty on suspect areas. Reference [12] compares the time efficiency of sequential survey against an algorithm which bases the survey pattern on uncertainty gradients of the readings. The information-maximizing approach of [12] is expanded to multi-platform systems in [16].

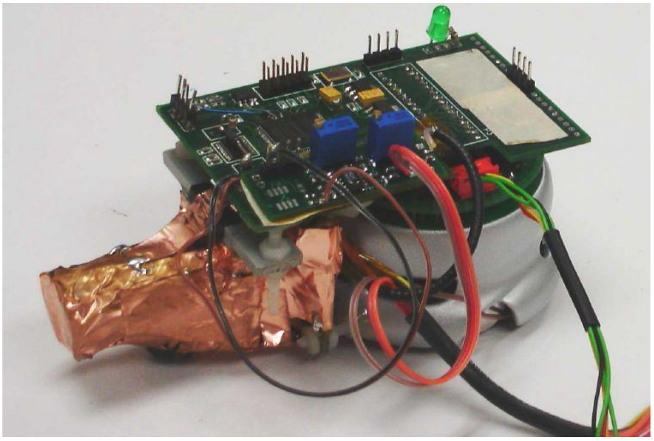


Figure 3-1: Kheperi II Robot with Alpha Counter Used by Cortez *et al.* [12]

Methods for measuring ground sources have been developed by Minamoto *et al.* [17] in response to the Fukushima accident. They account for the stochastic nature of radioactivity and use depth vision to estimate the distance from the radiation counter and the surface it is pointed at. This distance is used to estimate the source strength from the measured count rate based on simple geometric attenuation. The robot was teleoperated during this process, and the radiation counter was mounted on a pan-tilt unit so that it could be pointed towards a particular point.

Radiation surveying has been a function of robotics deployed in the field at the Three Mile Island, Chernobyl, and Fukushima Sites. The background of these deployments is given in Section 3.5. The radiation surveying tasks included taking measurements in the air as well as collecting scrapings and liquid samples for analysis.

In general, simple radiation counting with human control is well established. Future work should focus on automating the process and creating more complex analysis functionality. Cortez *et al*. [12] provide useful work on floor contamination search, especially with respect to controlling the drive speed in response to sensor data. It would be desirable to improve their demonstration with a more sophisticated heuristic for the sensor data and the ability to change the floor's mesh size in response to data. Minamoto's use of an articulated radiation counter would be useful for assaying specific containers. However, the simple geometric attenuation calculation may not be applicable to neutron radiation due to its high reflectivity. The most important new functionality needed for the vault application is generalized 3D mapping, with the ability to repeat measurements of specific points for the purpose of comparing changes to the flux profile over time. Automated spectrographic analysis for the purpose of species identification is also needed.

3.2 Object Identification

Kato and Billinghurst [18] developed a software package called Augmented Reality (AR) for use with video conferencing. The video conferencing involves participants wearing color cameras and 2-D barcodes to identify them to the software (Figure 3-1). The software identifies the 2-D barcodes using RGB vision. Each barcode encodes a unique number which can be associated with an object in a database. In addition, the software determines can determine the pose of the barcode relative to the camera. It is necessary for the barcodes to conform to the specification so that they can be correctly recognized. The interpreting software is supplied the size of the barcodes as a configuration parameter so that it can determine distance and pose. The patterns used on the barcodes are limited to those that are rotationally invariant.

Figure 3-1: AR Barcodes

Radio-Frequency Identification (RFID) technology can also be used for object identification. Robotic RFID integration has been tested using household objects. Passive, low cost tags can be used near-zero misidentifications and with no interference even with hundreds of tags in the environment. The main advantage over barcode identification is that line-of-sight is not necessary to detect an object. This would allow a robot to determine what containers are in a cabinet without opening it. The downside is that, due to the nature of electromagnetic permeation, it is much harder to determine the poses of the objects [19].

Geometric recognition can also contribute to object identification. Brian O'Neil and Adam Allevato have developed recognition software which classifies object features according to a new descriptor called a Cylindrical Histogram Projection (CHP). 3D range data recorded for the object, passed through the algorithm to produce a CPH, and then compared to a stored CPH for that object. The stored CPH's are produced using training data. Testing was performed using a set of objects typically found inside a nuclear glovebox, with recognition rates up to 97.2% [20] [21]. This method has recently been augmented with color recognition via RGB data, which will be documented in a forthcoming publication.

Barcode identification is the best current candidate for solving the problem. It fulfills the two most important criteria, which are accurate identification and accurate post estimation. The costs of implementing the barcode labeling are low. RFID is desirable for some tasks since it allows detection of containers without line of sight, but RFID is not currently approved for use in the facility. LANL is exploring the use of wireless technology however, so future work could involve combining these two technologies to achieve the benefits of both. Geometric identification should also be included to add an additional layer of verification by confirming that a container is the correct type and size.

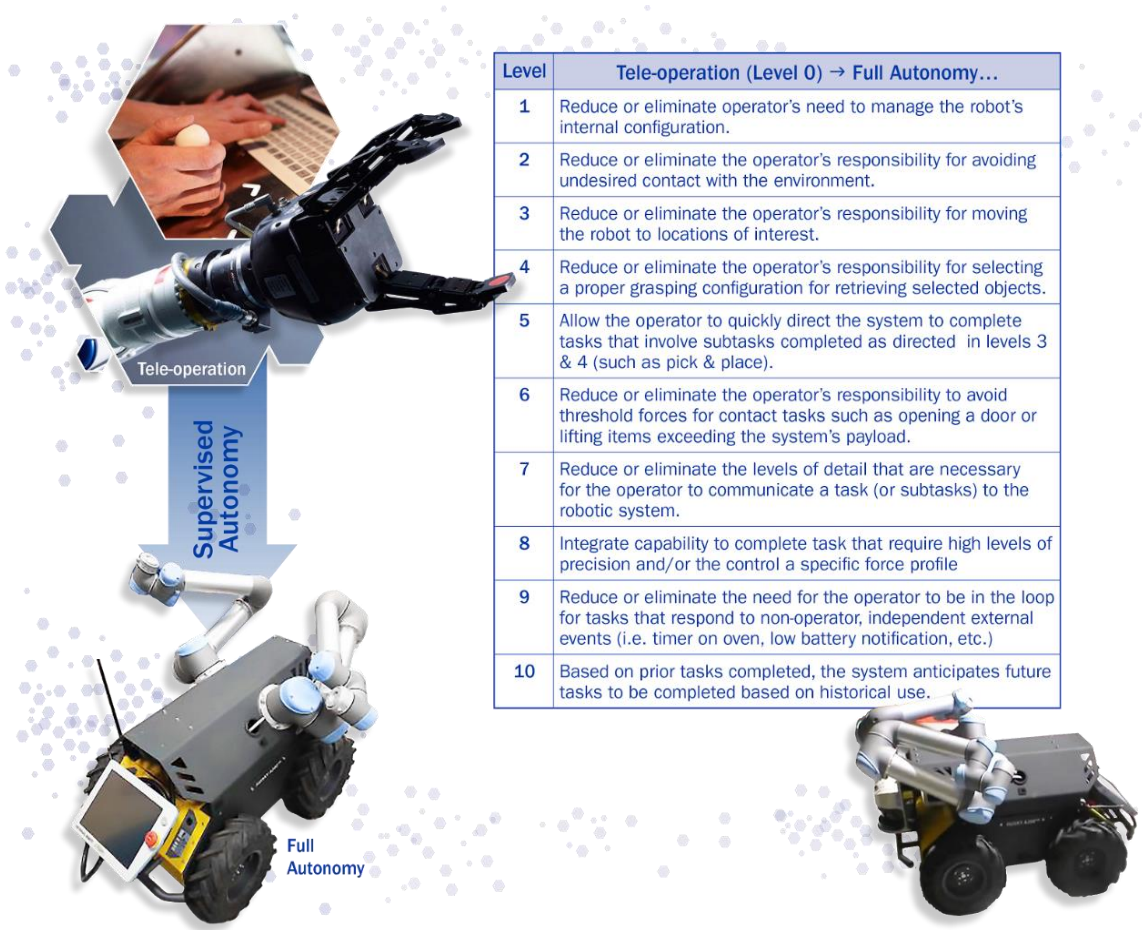
3.3 Levels of Autonomy/Safety Control

The UT Nuclear Robotics Group has performed prior work in layered models of autonomy [22]. Such a framework enables flexibility when dealing with a variety of tasks on a single system, since autonomy can be adjusted according to the difficulty of the task, consequences of failure, current availability of human labor, etc. Although LANL intends to use minimal autonomy until the reliability of the system is proven, variable automation is a topic for future work with incremental upgrades to the platform software.

A major benefit of variable autonomy is that it smoothes the transition into robotics for institutions that have no experience with them. Early on, robots can be kept on a "short leash" with total human control. As the operators become more experienced and facility procedures adjust to acknowledge the presence of machine labor, the robots can be allowed increasing autonomy to unlock their full potential. A spectrum of operating modes in displayed in Figure 3-2. Level 2 of the graph is expected to be the initial usage mode at LANL. However, the system can be upgraded with new capability after deployment.

A mobile system utilizing a variable autonomy framework was deployed at Idaho National Laboratory as discussed in Section 3.4.2. The same software was later installed on the NASA Robonaut with four operating modes configured [23]:

- In "Safe Mode," the platform will only take action to prevent the operator from colliding either the body or arms into an obstacle.
- In "Mixed Mode," mobility functions are automated while manipulation is stil manual.
- In "Shared Mode," mobility and grasping are both partially automated and respond to indirect control from the user.
- In "Autonomous Mode," the system is fully autonomous and the operator is removed from the loop.



Level	Tele-operation (Level 0) → Full Autonomy…
1	Reduce or eliminate operator's need to manage the robot's internal configuration.
2	Reduce or eliminate the operator's responsibility for avoiding undesired contact with the environment.
3	Reduce or eliminate the operator's responsibility for moving the robot to locations of interest.
4	Reduce or eliminate the operator's responsibility for selecting a proper grasping configuration for retrieving selected objects.
5	Allow the operator to quickly direct the system to complete tasks that involve subtasks completed as directed in levels 3 & 4 (such as pick & place).
6	Reduce or eliminate the operator's responsibility to avoid threshold forces for contact tasks such as opening a door or lifting items exceeding the system's payload.
7	Reduce or eliminate the levels of detail that are necessary for the operator to communicate a task (or subtasks) to the robotic system.
8	Integrate capability to complete task that require high levels of precision and/or the control a specific force profile
9	Reduce or eliminate the need for the operator to be in the loop for tasks that respond to non-operator, independent external events (i.e. timer on oven, low battery notification, etc.)
10	Based on prior tasks completed, the system anticipates future tasks to be completed based on historical use.

Figure 3-2: Transitional Levels of Autonomy [22]

Another important use of transitional autonomy is safety in co-robotic operation. When work alongside humans is planned, robots should be placed into autonomy modes that include force compliance safety features. For drive path planning, the industry literature provides work on human avoidance based on defining a "critical region" around a person. If the person and/or robot are moving, the algorithm computes the size of the critical region based on the current relative velocity and the acceleration capability of the platform. Testing demonstrated successful path-planning while avoiding collisions with human subjects performing erratic movements [24]. The NRG has also performed work in force compliance for robot manipulators which will be valuable in future work involving contact tasks [25].

3.4 Supervisory Control Techniques

Robotics looks to the generalized control systems field for techniques to manage the task execution of autonomous or semi-autonomous systems. The simplest method for robotic control is a linear script of commands to be performed in order. Such an approach is often adequate when dealing with straightforward tasks involving little uncertainty in well-defined workspaces. For more sophisticated applications however, it is desirable to organize tasks and behaviors into a structure. This improves organization, troubleshooting, and extensibility of source code.

Finite state machines (FSMs) are a widespread example of behavior organization, and have long been used in robotics. Continued research into state machines supports applications such as machine learning [26] and bipedal walking [27]. FSMs offer robustness due to their rigid nature, since action paths are derived from hardcoded state transitions. System behavior is highly deterministic in a given situation since it is all explicitly declared in advance by the designer. In robotics this confers advantages in terms of predictability and safety.

A more modern approach called Goal Oriented Action Planning (GOAP) is gaining acceptance in recent years. The drive for a more sophisticated framework is largely motivated by a need for more realistic yet elegant planners for artificially intelligent agents in video games [28] [29]. In this approach, the possible system actions are defined in terms of pre-conditions and post-conditions. This greatly simplifies the designer's task, since it replaces hardcoded behavior with procedurally generated task execution. With proper definition of the worldstate variables and system actions, GOAP can envelop the entire state-space. Since actions are not coupled to each other as states are in a FSM, it is much simpler to add new behavior to the system. Therefore the designer can implement new behavior with a minimum of re-coding. It is also convenient when dealing with multi-agent systems, in which two agents might have the same goal but different action sets. Sending a command to the system is as simple as defining a desired worldstate and submitting it to the algorithm.

There are also multiple means of defining actions in the world space of the action planner. The Stanford Research Institute Problem Solver (STRIPS) is an established method of specifying the planner inputs [30]. Conditions and actions in STRIPS are predicates of objects. For example, the IsAtLocation (item,location) condition would provide a true/false result for the entire space of items and locations. Actions are specified on the executable level. By contrast, Hierarchical Task-Network (HTN) planning is another approach which organizes actions into networks of complex tasks built from other subtasks [31]. Both approaches are often used as the basis for GOAP. The claimed benefit of HTN is that the ability to define compound tasks allows for more natural action specification. The downside is that it introduces some action coupling back into the GOAP algorithm, which may make design changes more complicated.

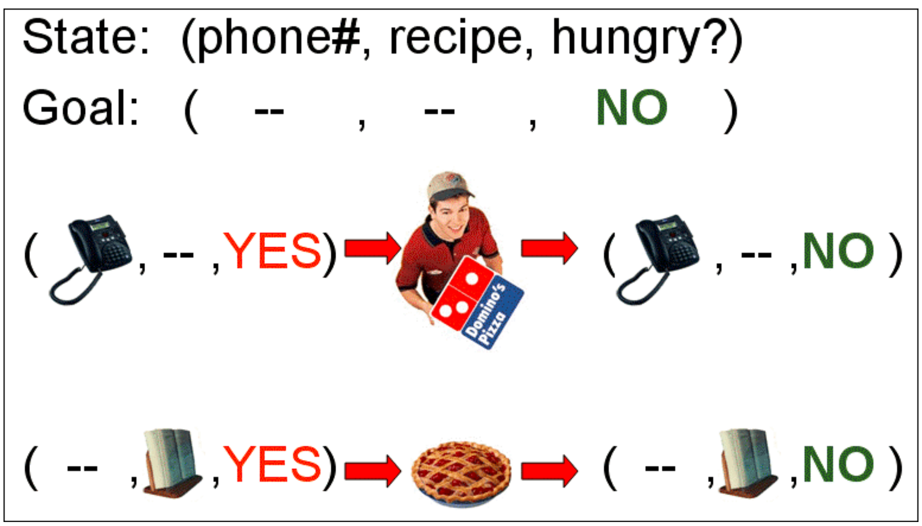


Figure 3-3: An Example Goal Oriented Action Plan [28]

3.5 Past Field Deployments in Nuclear Environments

3.5.1 Early Efforts

Remote robotics is nearly as old as the field of nuclear engineering itself. The need to handle spent fuel in plutonium separation processes motivated the first telemanipulation system, the Master-Slave Manipulator Mk. 8 (Figure 3-4). Hot cell requirements have continued to be a primary driver of robotic grasping technology, with descendants of the Mk 8 present in all high-radiation laboratories. These devices have achieved great success because they work in tightly defined operating spaces with total human supervision.

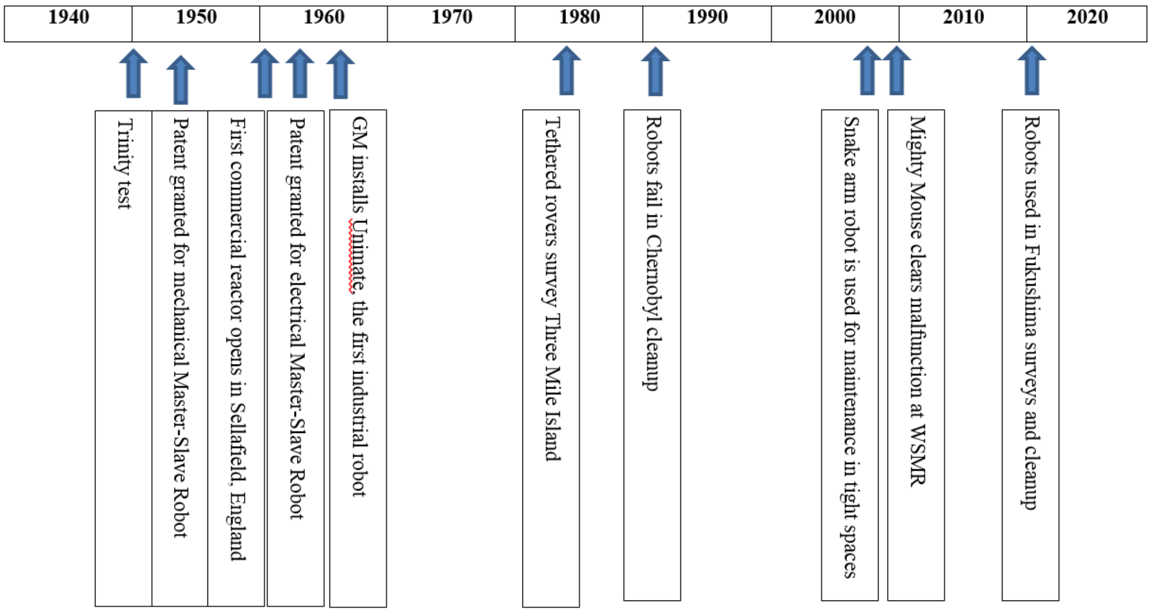


Figure 3-3: Milestones in the History of Nuclear Robotics

Figure 3-4: Master-Slave Manipulator Mk 8 [32]

Mobile robotics has a younger and rockier track record in the nuclear field. Accident response has been the main driver, with primitive systems deployed at the Three Mile Island (TMI) and Chernobyl sites. Two platforms were sent into TMI during the 1980's. The first, called Rover, successfully performed tasks such as core sample drilling, visual inspection, and radiation measurement. The system was rugged but simple; for example, rather than receiving radiation readings electronically, operators had to point the cameras at a dial display on the instrument. After a few years of operation the Rover was retired and left inside the structure due to contamination. The second system, Workhorse, was much more ambitious but ultimately too complex for the intended environment [33].

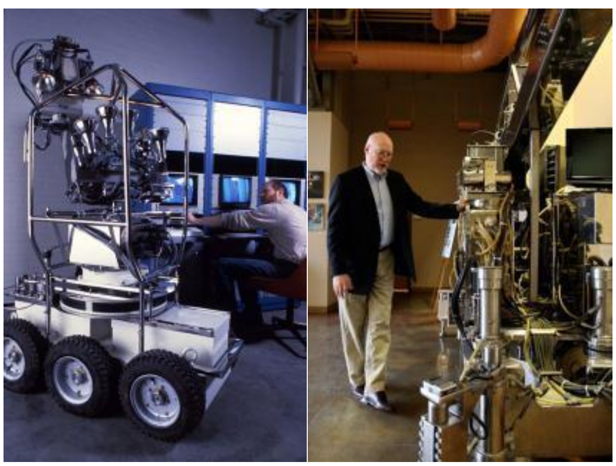


Figure 3-5: Rover (Left) and Workhorse (Right)

Dozens of robots of widely varying designs have been used at the Chernobyl site. In the immediate aftermath of the disaster, mobile platforms were used to push debris off the damaged roof of the reactor building. However, these robots failed within a couple days due to radiation damage on the electronics and the authorities were forced to rely on human labor [34]. Since then, a long line of machines have been used for purposes ranging from visual inspection to radiation measurement and sample collection.

3.5.2 Idaho National Laboratory

Idaho National Laboratory (INL) performed a real-world test of a mobile radiation survey robot [34]. Operators used a teleoperated system to survey a heavily contaminated decommissioned pump facility. The resulting human radiation exposure was reduced by over 90% compared to baseline operations. More labor was required to deploy the robot than the baseline operation, but many more data points were collected. The system also featured a variable autonomy system along the lines of Section 3.3.

The main difficulty encountered was in wireless communication, as the thick reinforced concrete walls of the facility blocked the wireless signal is some areas. Tethered communication was necessary to operate in those cases. The platform was also not capable of navigating the stairways, so operators had to carry it between floors. Operators also entered the facility to install wall cameras in the rooms that the robot would be working in. Despite this, total human exposure was greatly reduced since it was not necessary for them to approach the contaminated equipment to be surveyed. Therefore, this deployment demonstrates the improved worker safety and data collection that robotic systems can provide in radiation surveying tasks.

3.5.3 Fukushima Daiishi

Six missions were performed at the Fukushima disaster site using a teleoperated mobile robot [13]. The robot completed visual inspection, radiation dose measurement, and air temperature measurement tasks. The main difficulties were caused by debris and tight spaces impeding navigation. In the sixth and final mission, the communications tether became snagged on piping and the robot could not be recovered. The report provides valuable experience on teleoperated control, radiation survival of digital electronics, and tethered communication.

3.5.4 Savannah River Site (H-Canyon)

A series of robot were deployed to the H-Canyon facility at Savannah River Site. H-Canyon is a decommissioned plutonium separation plant. The goal of the deployments was to inspect the condition of an air exhaust tunnel used to maintain negative pressure in the facility. This simply involves visual inspection using a video camera mounted on an elevation arm and pan-tilt-zoom unit. The development of these robots gives guidance on the use of tethers. Before deployment, the designers tested the robot's pull strength and ability to pull the tether around a sharp corner. They also developed the ability to lower the robot down a 10 foot [3 meter] ledge using the tether and ensure that the robot rights itself at the bottom. Kevlar strain relief was incorporated into the tether of the third robot after the tether of the second robot failed due to tensile stress during the ledge drop [35].

3.5.5 Lessons Learned from Previous Efforts

The major obstacle faced in these efforts was mobility. Robots often failed to navigate spaces with even small amounts of debris, and caught tethers were a recurring problem. Stairs are especially difficult yet common in the large facilities being surveyed. Improving operator awareness is key to dealing with these problems as driving with limited vision increases the odds of getting stuck on an unseen protrusion. Treaded robots have been shown to generally deal with debris and stairs much better than wheeled robots, and shorter robots can turn in tight spaces more easily.

Wireless communication is preferred over tethered, since cables can become snagged and immobilize the robot. However, wireless communication is often not possible due to signal dead-zones caused by thick walls. Tethers should be fitted with strain relief and be securely connected at the robot end. The robot also needs to be tested with respect to pulling the tether around corners.

3.6 Inventory Robotics Research in Academia

An inventory inspection robot was designed by an undergraduate team at Calvin College for their senior design project. The purpose of the robot is to check that books on a library shelf are sorted correctly. Although the full system could not be built on their budget, they were able to perform rudimentary integration testing. The books were identified using passive RFID tags and traditional barcodes. The full robot can read books at different highs using a vertical prismatic joint. The robot can autonomously inspect one side of an aisle after being positioned by a human, but will not move to new aisles on its own. The only navigation sensing is a forward proximity sensor to stop it from driving into obstacles [36].

3.7 Commercial Robots for Inventory/Warehouse Applications

There are a variety of commercial robotic systems on the market for managing inventory and performing operations in warehouses. PAL Robotics offers the Stockbot, a 1.9 meter tall autonomous platform which detects RFID tags on merchandise. It is intended to replace the normal retail practice of having employees individually scan every item with a handheld barcode reader. It uses depth vision based mapping, localization and collision detection. The operator creates a map using teleoperation and later can define a portion of it for the platform to sweep autonomously. This readily demonstrates the mapping and navigation aspects needed for the vault application, and shows the usefulness of RFID object identification [36]. Keonn Technologies also offers an RFID based inventory robot that is virtually identical in form and function to the PAL product [37].

Figure 3-5: PAL Robotics Stockbot (Left) and Amazon Drive Units (Right)

Amazon Robotics (formerly Kiva Systems) provides what is probably the most well-known example of mobile warehouse robotics. The Amazon system involves a swarm of centrally controlled robots (called drive units) that can navigate to inventory locations and retrieve entire cabinets or pallets of items. Drive units are allocated to tasks by a central coordination and dispatching system that tracks the state and location of all the units. Units navigate by scanning 2D barcodes arranged on the floor in a 1m x 1m grid. Units have collision avoidance routines and can react to interrupt conditions such as needing to recharge. This demonstrates complex task planning and execution (since robots must be coordinated to avoid each other), as well as using 2D barcodes for localization purposes [38].

Fetch Robotics produces a similar system called Freight, in which mobile platforms use manipulators to pull items from shelves instead of moving the entire shelf. They also use depth vision navigation along the same lines as the NRG platforms [39].

A radically different solution is the AutoStore, in which inventory is condensed into a fixed grid-like structure. The robots drive on top following tracks. This solves the problem by essentially removing all uncertainty from the workspace of the robots [40].

Clearpath Robotics has released a video of a prototype mobile manipulation system using a Ridgeback platform and UR5 manipulator. The demo video shows it retrieving an item from a warehouse location, but does not clearly show if this is done autonomously [41].

Chapter 4: Software Component Development

Based on the technical challenges listed in Chapter 2, it was necessary to develop new capabilities on the NRG mobile platform, including:

- Interfacing with radiation counters and performing basic radiation statistical analysis.
- Visually presenting radiation data.
- Using a remote inventory database to facilitate inventory activities.
- Querying and modifying an inventory database.
- Procedurally generating action plans to accomplish complex goals.

This chapter details the new software that was written to implement these functions. NRG software is written using Robot Operating System (ROS), which is described in the following section.

4.1 ROS Overview

All software described in this chapter has been written using Robot Operating System (ROS). ROS is an open source framework created by the Open Source Robotics Foundation for developing robotics code. ROS is developed to run on Unix based systems, with Ubuntu Linux as the officially supported distribution. The core feature of ROS is that software is organized into "nodes," or self-contained functional components. Nodes communicate with each other by broadcasting their output over topics which can be read by other nodes, or by responding to service requests from a specific node. The ROS master node launches the other nodes, sets up the required topics and service advertisements, and manages the data travelling over them. Data are packaged in ROS Messages which are transmitted between nodes via TCP/IP. The ROS standard library provides many message types useful for robotic applications, and users can define their own message structures as well [42]. One or more related nodes and their supporting files are encapsulated in a "package."

ROS supports both C++ and Python, with the core library offered in both languages. Nodes written in both languages can be run simultaneously and pass ROS Messages to each other. Unless stated otherwise all code in this chapter uses the Jade Turtle release of ROS.

The key advantage of ROS is modularity. The nodal architecture of the code allows development to be broken down per a black-box philosophy. Individual nodes are responsible for specific functions of the overall program, allowing code and data to be compartmentalized. This simplifies collaboration between team members since each person can take responsibility for different nodes. It also helps with troubleshooting since developers can diagnose bugs by inspecting the message traffic between nodes and then determining which node is transmitting incorrect results.

The nodal structure also simplifies integration of different hardware. Sensor, drive bases, manipulators, grippers, etc. can be handled by different driver nodes. Nodes can also be distributed across a network of computers. Some nodes are run locally on the platform processor and others on the remote workstation. More computers can be added as necessary to increase the processing power of the network as more demanding nodes are added to the overall program. It is beneficial to cluster nodes that exchange large amounts of data with each other on the same machine. This reduces the bandwidth load on the network. The most common example is to run vision processing nodes on the platform.

4.2 Radiation Sensing

4.2.1 Vault Application Requirements

The system is intended to replace human Radiation Control Technicians (RCTs) in several tasks. First, the vault operators are required to perform a radiation survey of the entire vault every quarter. This involves taking a series of radiation doses rate measurements at waist height in a 1m x 1m grid pattern. The radiation instruments need to be able to measure dose rates on the order of hundreds of mrem (or on the order of mSv) per hour. Currently this is done using hand instruments and estimating the current position. These measurements are dominated by neutron radiation with a much smaller gamma component. These emissions do not experience significant absorption in air over typical distances in the vault. They are capable of penetrating the walls of the containers, but the thick cement walls of the vault significantly attenuate them. The purpose of the robotic system is to produce this data with greater accuracy and repeatability in positioning while eliminating the dosage to the RCT.

The system should also be capable of performing the periodic floor contamination sweeps. Historically, floor contamination searches have been performed using a wide variety of equipment. Broom-like instruments with very large counting areas are common. The focus is on alpha contamination such as uranium and plutonium oxides or americium. Alpha emissions have a typical range of only a couple inches and cannot penetrate the outer skin. Alpha sources can cause major injury if ingested or breathed in, however. For non-contact contamination search, the lab stipulates a counting distance of 0.25 inches and a sweep rate of 2 cm/s [1].

The preferred method, where practical, is to swipe the area with a swab and then take counts from it instead of directly off the surface. This is because the real concern is "removable" contamination, which can get on clothing and hands and cause an uptake. Non-removable alpha contamination is not much of a danger. Swipes are generally not used for floor sweeps since the areas involved are too large, but would be necessary for surveying cabinets or containers. For this report, we focus only on the non-contact method.

The retrieval procedure for containers includes a non-contact radiation assay after its removal from the vault. A dose rate measurement is taken at a distance of 30 cm.

4.2.2 Radiation Counting Statistics

The number of decays that a radioactive sample will undergo in a given time is not deterministic even when the activity is known. Each individual atom in the sample has a 50% chance of decay in every half-life. In general the decay probability in any given period of time given by the decay constant:

$$\lambda = \frac{\ln(2)}{t_{1/2}},$$

where $t_{1/2}$ is the half-life of the sample. Radiation statistics makes use of the Poisson distribution to represent the likelihood that a given number of decays will occur in a certain period of time. The Poisson distribution is a discrete probability distribution defined as:

$$P(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!},$$

where x is the number of decays.

The Poisson distribution has the following properties which make it appropriate for modeling radiation [43] [44].

- The distribution is binomial in that each atom in the sample has exactly two possible states: decayed or not decayed.
- The expected value of the distribution is equal to the decay constant λ, so that if x is the number of decays and t is the elapsed counting time:

$$\lambda = \lim_{t \to \infty} \frac{x}{t}$$

- The variance is equal to the discrete variable (the number of counts), and therefore the standard deviation is the square root of the counts. Practically, this means that in order to reduce the uncertainty of a radiation measurement by a certain factor, the counting time must be increased by that factor squared:

$$Uncertainty \propto \frac{1}{t^2}$$

- The individual events (decays) are statistically independent. In other words, each atom's likelihood of decaying is solely dependent on its decay constant and is not influenced by the state of other atoms.
- The likelihood of an event is independent from the number of events that have already taken place. As a practical constraint, this means that the total number of atoms must not appreciably change during the counting period.

The radiation detection functions of the system must account for the probabilistic nature of radioactivity. Probabilistic considerations take on considerable importance in applications involving low-level counting, such as searching for small amounts of contamination on surfaces or counting long half-life species (which most uranium and plutonium nuclides are). In general, for any given sample there will be a minimum measurement time needed to ensure that it achieves the desired degree of confidence. For a Poisson distribution, the coverage factor is given by:

$$Coverage\ Factor = \frac{x}{std.dev.(x)} = \frac{x}{\sqrt{x}} = \sqrt{x}$$

The coverage factors associated with commonly used confidence level, for a confidence interval symmetrical about the mean, are shown in Table 4-1:

Coverage Factor	**Area Within Confidence Interval (%)**
1.0	68.3
1.645	90.0
1.96	95.0
2.576	99.0
3.0	99.9

Table 4-1: Coverage Factors and Associated Degrees of Confidence [44]

Again, this shows that increasing the counting time to increase the measurement certainty is subject to diminishing returns.

4.2.3 Software Implementation

A ROS package called `radiation_sensing` was created to contain software related to measurement and analysis of radiation. The base level node, called `radcount_action_server`, is used to control a simple counting instrument, perform some basic statistical analysis, and package the data into ROS messages for use by the rest of the system. The node is designed for an instrument that simply sends a voltage pulse every time a count is detected, and the functions are encapsulated in a ROS action server. When the action server is called by a client node, it begins listening to the instrument on its "count" topic. Every time a count is received - or at least every five seconds – a feedback message is returned to the client providing the cumulative count, average count rate, and Poisson upper and lower limits.

The ROS action can be used for source detection, in which the reading at the present instrument location is compared against a fixed background. The node will gather counts at that location long enough to reach the confidence limit specified by the client. Once the lower confidence limit exceeds the specified background threshold, the system will declare that the counts are statistically significant and that a radiation source is present near the survey location. Likewise, if the upper confidence limit falls below the threshold, the system will declare that a radiation source is not detected.

Other nodes in the `radiation_sensing` package implement more complex measurement functionality. Node `rad_point_survey` encapsulates a drive command with a call to `radcount_action_server`, so that the robot will perform a radiation survey of a specific point in the world frame. Node `rad_survey` provides a text user interface for general surveying. The user can activate manual survey mode to teleoperate the robot to desired points and start/stop the measurements. The user can also command an autonomous survey, performed with a series of calls to `rad_point_survey` in order to survey multiple points. The autonomous survey will pull the requested waypoints from a specified waypoint file, which can also be created by the user during manual mode. A known map of the survey area is required for autonomous mode. These can also be created during manual mode for later use.

Node `rad_survey` also includes functions for storing localized radiation readings. Later surveys can repeat the original survey pattern and make comparisons between past and present data. A robot demonstration of this program is discussed in Section 5.6.2.

The software is based on the assumptions inherent to the Poisson distribution discussed in the previous chapter, since they are valid for the intended vault application. The materials of interest are plutonium or uranium in various forms. These species have sufficiently long half-lives that an appreciable portion of the atoms will not decay during a counting period. These species will also not be produced from any parent nuclei during the count.

4.3 Barcode Sensing

The system utilizes the `ar_track_`alvar [18] package for 2D barcode marker recognition. The inventory management tasks use these barcodes to identify items of interest in the environment such as material containers. The `ar_track_alvar` algorithm uses RGB vision to recognize the format specified for the marker, and to identify the number encoded on the marker. It uses depth and RGB vision to determine the pose of the marker relative to the camera, so the location of the item it encodes is known to the system. This means that the marker codes are limited to patterns that are rotationally invariant to prevent ambiguity.

A ROS package called `barcode_recognition` was written which receives recognition messages from `ar_track_alvar` [45], which is a ROS wrapper of the open source marker recognition package ALVAR [46]. The node `barcode_recognition` performs additional processing and broadcasts the results to the system. `barcode_recognition` can perform frame transformation to find the barcode poses relative to the world map or various frames of the robot. It can also interface with a ROS visualization node called RVIZ (discussed in more detail in section 5.4). When barcodes are detected, `barcode_recognition` constructs and publishes a message which instructs RVIZ how to visually represent the detected barcodes. The barcodes will be visualized at their 3D pose as determined by `ar_track_alvar`, including any spatial transformations performed by `barcode _recognition`.

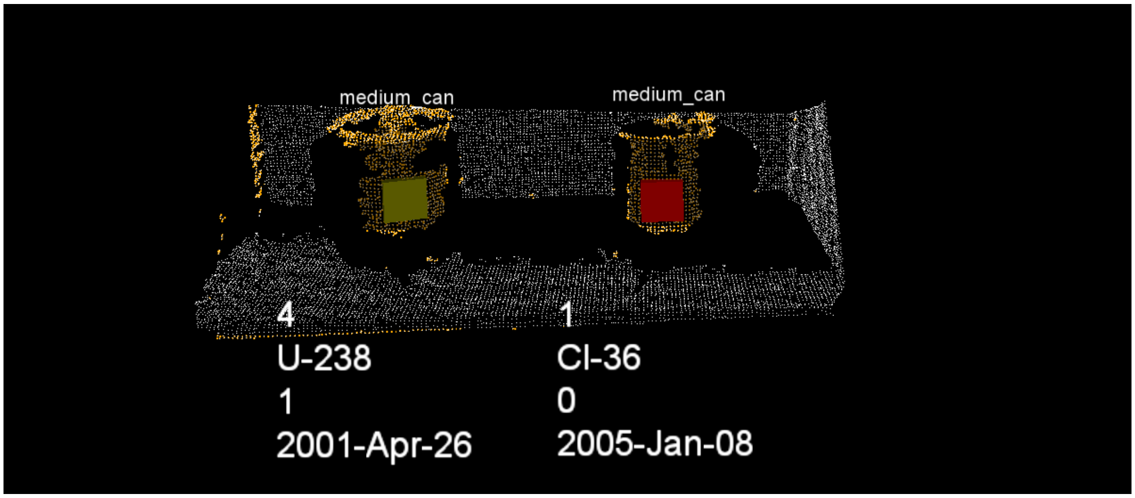


Figure 4-1: RVIZ Visualization of Barcode Poses

The `barcode_recognition` node also interacts with the inventory database node discussed in section 4.5. When a barcode is read, it will request a database query to determine if the value of the barcode represents a container. If it does represent a container, `barcode_recognition` will receive data associated with that container (contents, activity, etc). It will then construct another visualization message for RVIZ which contains that data organized into a string. This will cause the data to be displayed in RVIZ below the marker for the barcode pose. This allows an operator to instantly see the details of a container as soon as it is detected by the robot.

4.3.1 Barcode Marker Requirements and Limitations

A barcode consists of a white background, black border, and black-and-white interior grid. The border and background are necessary to define the four corners of the marker and thus provide reference points to the image parsing algorithm. The algorithm uses the known side length of the square border to perform a perspective transformation and determine the distance and orientation of the marker.

The black border around the marker cannot be occluded. Any break in the border edges will cause the recognition to fail. Another open source marker recognition package called ARTag claims to have solved this limitation [47], but it is not currently packaged for ROS. The barcode value is encoded in the black-and-white interior grid. Some bits are fixed and serve as a reference for determining orientation.

The number of bits in the grid can be chosen by the user (currently 5x5 for the barcodes in NRG demonstrations). Increasing the number of bits increases the number of values that can be encoded, but makes distinguishing different markers more difficult. Thus there is a trade-off between range and barcode resolution. In the vault application, the system should rarely need to recognize container barcodes more than about 3 feet (1m) away, so a high resolution is permissible. Barcodes used for robot localization should use a lower resolution and/or larger marker size to allow greater range, however. Fortunately, a large number of different localization barcode values should not be necessary. The number of possible values for a given resolution is given by the following formula and Table 4.2.

$$P = 2^{(n^2-2)} - 2^{\left(\frac{2n^2-(-1)^n-7}{4}\right)}$$

where P is the number of permutations and n is the number of pixels in a side.

Resolution	Permutations
2x2	3
3x3	120
4x4	16,320
5x5	8,386,560

Table 4-2: Permutations for Different Barcode Resolutions

A program is provided in the ALVAR package for generating images files of valid barcodes. These can then be printed out and affixed to objects.

Testing was performed by a University of Texas senior design team to determine real world performance of the detection algorithm. Barcodes with dimensions of 4.5 cm x 4.5 cm and 5x5 resolutions were used. The vision system was an Asus Xtion Pro [48] depth sensor mounted on a tripod. Figure 4-2 shows the dimensions for measuring the performance. With the markers pointed directly towards the camera and directly in front of the camera (θ=0°), the maximum detection range was found to be D=40 inches. The maximum allowable tilt angle between the camera and barcode was found to be θ=35 degrees, and the maximum range at this angle was D=37 inches. It was also found that motion of the camera reduces the recognition rate (measured as the percentage of frames in which the barcode was detected) [49]. Therefore the deployed system should be kept stationary when accuracy is needed.

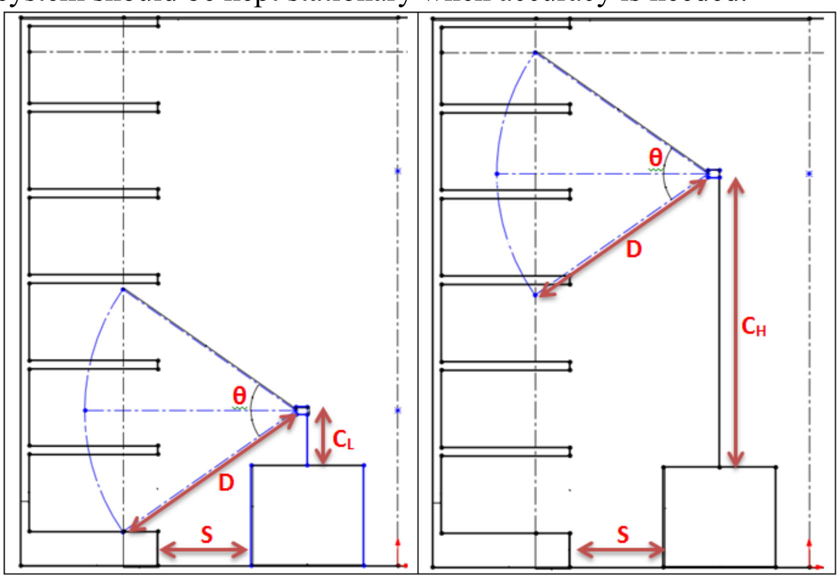


Figure 4-2: Dimensions for Barcode Performance, S = 18

4.4 Localization/Navigation

At first the system is intended for teleoperation, but eventually it will require the ability to autonomously navigate the vault space. It must be capable of localizing itself at the entrance as well as re-localizing at any point inside (in case of faults during a task). Furthermore, independent navigation and localization capabilities provide additional safety features that can assure damage from operator errors is prevented.

The robot accomplishes autonomous navigation in mapped areas by using the robot sensor suite to determine its location and track its motion through the mapspace [50]. The prototype possesses a 2D IR rangefinder and an Inertial Mass Unit (IMU) for navigation.

During the localization phase, a distribution of pose “beliefs” of equal probability is populated on the map. As the robot moves, LIDAR data are collected and a Bayes filter uses them to iteratively adjust the belief distribution until it converges on the true robot position (Fig. X). The equations for a single iteration step are shown below:

Prediction of posterior distribution from prior state and control input u:

$$bel(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$

Correction of distribution using sensor data set z:

$$bel(x_t) = \eta \, p(z_t|x_t) bel(x_t)$$

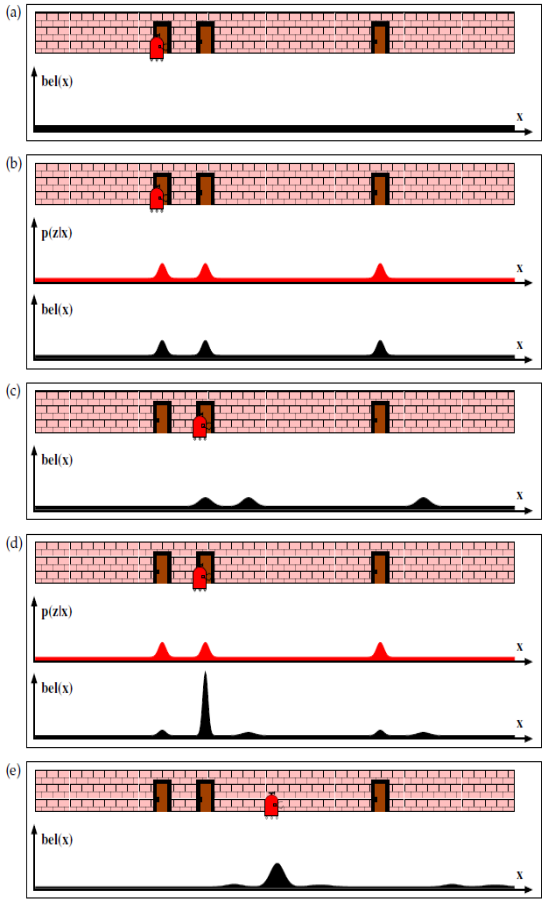


Figure 4-3: Bayesian Robot Localization [50]

p(z|x) is the probability of detecting a door (measurement z) at a given location x, including probabilities of false positives and negatives.
bel(x) is the probability of the robot being present at x.

Following localization, navigation to target poses is accomplished using Dijkstra's Graph Search Algorithm, which computes a shortest-path solution [51]. The platform can avoid unexpected obstacles on the path, including mobile obstacles. This algorithm provides robust navigation performance despite sensor uncertainty and unmodeled dynamics such as wheel slip. Reference [50] provides a good overview of probabilistic analysis as it related to robotic (particularly mobile) applications.

Using Bayes' Rule for navigation is well established, but it has additionally been applied to robotic perception and pose estimation [50], Condition-Based Maintenance applications [52], and machine learning [53]. It can also be used to address other technical issues of the Vaultbot including radiation sensing and label recognition, and it provides a mechanism to properly evaluate their combined import related to tasks in the vault including radiation surveys and inventory.

4.5 Autonomous Task Planning

Task automation requires that the system be able to understand a complex instruction from the user and decompose it into a sequence of basic operations. For example, a command to retrieve a container from the vault can be decomposed into:

1. Query database for container location.
2. Travel to container location.
3. Open cabinet/drawer.
4. Verify container identity.
5. Grasp container.
6. Close cabinet/drawer.
7. Return to vault entrance.

These tasks could be broken down into even more basic steps. Traditionally, this kind of task execution has been organized using a state machine architecture, in which a behavior and state transition is explicitly defined by the system designer for every possible state. However, this approach quickly becomes untenable when dealing with increasingly large sets of state variables. It is also time consuming to add new behavior to the system, since the system designer must create new transitions between any new states and the existing states.

GOAP (see section 3.4) is a planning procedure in which the system is defined in terms of actions, unlike a classic state machine which defines a system in terms of states. As in a state machine, the world is modeled as a collection of Boolean state variables. However, the discrete world states are not associated with any particular behavior. Rather, behavior is associated with the actions themselves, which are defined by their state variable preconditions and their effects on the state variables (postconditions).

Cost functions for the actions can also be defined. When given a task, the system will perform an A* graph search (see Section 4.5.4) to find the lowest cost action path from the current worldstate to the specified goal worldstate. An "action stack" for the system is built that traverses this path. The system can then pull actions off the stack one at a time, running the function associated with each one. If an action fails or if a state variable changes unexpectedly (for example, variable `contamination_detected` becomes true), then the system can stop and call the A* algorithm to generate a new plan which accounts for the new world condition.

4.5.3 Code Examination

A ROS package called `task_planning` has been created for use with the NRG platforms which contains a C++ implementation of GOAP. The code manages the worldspace variables, performs the A* graph search, and returns the action stack along with predicted cost of the plan. The worldstates and actions can be defined and imported using JSON files, which lets operators make changes to the robot behavior without editing source code.

The GOAP implementation is built around the following data structures. Portions of this code are derived from an earlier action planning package by Abraham Stolk [54].

worldstate_t – Represents a set of state variables, called **atoms**. Can be used to represent a goal or track the current surroundings of the robot. The Boolean values of the state variables are modeled as a `boost::dynamic_bitset`, which allows a Boolean to be compressed into a single bit rather than a whole byte. It also supports bitwise logic operators which greatly optimizes the GOAP algorithm. This object allows dynamic resizing so that modifying behavior by adding more atoms is simple. The structure also has a second `boost::dynamic_bitset` member for tracking which atoms are known and relevant. This is necessary since a goal state might not care about certain atoms in the worldspace.

action_t – Represents an action which modifies a world state. It has `worldstate_t` members to represent the preconditions and postconditions. It also has a function pointer member which can be referenced to a cost function for the action. The cost function is of the form:

```
float CostFunction(actionplanner_t*, worldstate_t*);
```

Since the second parameter is a `worldstate_t` instance, the cost may depend on the situation the robot is in. `action_t` also has a float member that can store a constant cost value if a cost function is not needed. A member method `get_cost` will return the action cost, using the cost function if one is assigned and the cost data member if it is not.

actionplanner_t – Represents all the actions in a worldspace. Members include a `std::vector<action_t>` to hold the actions and a `std::vector<std::string>` to hold the names of the actions.

A class called GOAP encapsulates the API functions with which a user of the library can manipulate these structures. This includes adding atoms to worldstates and setting their values, defining actions and assigning their cost functions, and calling the planner function.

4.5.4 A Search*

The A* search algorithm is a common graph traversal method for finding lowest cost routes between two points in a discretized space. It improves upon the original Dijkstra algorithm [51] by incorporating a heuristic function. A well-chosen heuristic will cause the search algorithm to converge on a lowest-cost path faster than the naïve search of the original algorithm. The heuristic is of the form:

$$f(n) = g(n) + h(n)$$

where f(n) is the total cost of node n, g is the cost of going from the initial state to n (the posterior cost), and h is the estimated cost of going from n to the goal state (the prospective cost). Note that h(n) is merely an estimate; knowing the true value would require running the rest of the graph search for n, which would reduce this back to the naïve search algorithm. The magnitude of the heuristic relative to g(n) can be adjusted to make the search algorithm more or less greedy. A large heuristic will make the algorithm greedier, stopping early even if the chosen path might not be optimal. A small heuristic will cause the algorithm to spend more time searching for an optimal path.

The A* search in this implementation uses the following heuristic:

$$h \;=\; C * n(A \backslash G),$$

where h is the minimum cost from node A to the goal G. In other words, h is the number of relevant state variables that are different between node A and goal G. C is a multiplier constant which can be used to tune the greediness of the graph search. A larger value will increase the heuristic and the greediness. During the current NRG demonstrations a small value for C (0.1) is used since the worldspace is not large enough to make processing time a limitation.

A possible optimization of the algorithm is to use a heuristic that obeys the following:

$$h(a) \leq g(b) + h(b)$$

for any two nodes a and b. This means that a node does not need to be run more than once, and can be placed in a "closed" set after the first iteration. Such a heuristic is called *monotonic*. The heuristic used in this implementation is not monotonic, however. Closed nodes can be re-examined and have their cost reduced.

The time complexity of A* in Big O notation is given by

$$T(b, d) = O(b^d),$$

where b is the average number of transitions per state, and d is the length of the shortest path [55]. This means that, in general, planning will take longer for tasks that require more actions and for worldspaces in which the allowable actions of the system are not highly constrained by preconditions.

4.6 Database Interface

A ROS package called `inventory_mgmt` has been written to implement inventory related operations. It includes a node called `inventory_database_interface` which manages a connection to a SQL database to facilitate this.

The `inventory_database_interface` node provides ROS services to:

- Connect to and disconnect from the database.
- Query the database for a container specified by its `container_id`. Node `inventory_database_interface` pulls the associated data fields for that container and sends them to the client node.
- Query the database for a specific inventory location. Node `inventory_database_interface` pulls a list of all containers whose `location_id` matches the provided key and then sends the list of those `container_id`s to the client node.
- Determines what containers, if any, should be in the robot's field-of-view given its camera pose. First, the function queries all containers in the database whose `location_id` matches the provided room (it is assumed that containers in other rooms cannot be visible). It then uses the camera's field of view and range to perform a frustum culling on those containers' poses. Finally, it accounts for occlusion by getting a depth

map from the camera and transforming the containers' poses into the depth map for comparison. Containers with greater depth than the associated pixel of the depth map are deemed to be occluded and are not included in the result.

The interface node is written to be compatible with the PostgreSQL version 9.3 implementation of the SQL Standard [56]. Connections are made with a username and password using MD5 hashing. The node can be run on a different machine other than the database server; however the database server must be configured to allow remote connections.

A pair of C++ classes, Container and Location, is used to define the fields associated with the database entries. Query operations to the database will return a vector of these classes containing the resulting data. The classes are made available for use in other ROS code via a C++ library. A Container object supports the following data fields and types:

Field	**Data Type**	**Description**
container_id	integer	Unique identifier of the container. Primary key of the database table.
contents	string	Contents of the container.
activity	real	Decay activity of the contents at time of deposition.
date	string	Date of the material, for use in decay computations.
pose	real[7]	6-DOF pose in the world frame.
location_id	integer	Unique identifier of the inventory location. Foreign key from the database table.

Table 4-3: Container Database Fields.

Currently the Locations table is simply a list of unique `location_id` entries. Future work includes adding navigable poses to the Locations table. This will allow the robot to look up navigation goal poses when ordered to drive to inventory locations, rather than relying on hardcoded poses.

Chapter 5: Mobile platform integration

This chapter describes integration and testing of the software packages described in Chapter 4. This work involves demonstrations on the NRG mobile platforms as well as benchtop tests. The focus is on the Pioneer LX since it is the newest platform, it was used for the most recent developments, and it is the intended platform for work in the near future.

5.1 Platform Hardware Overview

The most recent integration work has used the Pioneer LX system, described in Section 1.3.3. This consists of installing the software packages detailed in Chapter 4 using the practical setup and modular testing methods described in this chapter.

The Pioneer has several built-in sensor systems such as forward IR rangefinders for navigation, forward and rear ultrasonics for proximity detection, and bumpers for collision detection. NRG augmented the system with additional accessories such as an Asus RGB and depth vision system and alpha radiation detector. The hardware was also upgraded by adding an auxiliary mini-PC for dedicated vision processing [57].

Future hardware integration work includes installing a ZipperMast system. The ZipperMast is a motor driven extension mast which can be used to elevate hardware up to 48 inches. This is valuable in the intended vault application since it will be necessary to inspect items that are higher than the robot base. The ZipperMast works by unspooling three stainless steel bands which interlock in a triangular mast, becoming rigid in the process. This system offers an exceptional length of extension compared to its non-extended volume. NRG plans to mount vision and radiation sensors on the top of the mast to enable inspection of objects about the ground level.

5.2 Vision

The system is equipped with an Asus Xtion Pro vision system. This includes an RGB video camera and a laser rangefinder array for producing depth images. The Asus is mounted on the front of the platform and angled downward for viewing objects directly in front of the robot. The system can transmit an RGB feed from the Asus to the operator workstation. RVIZ can be used to display this feed and apply various data overlays such as barcode markers and depth images.

The ROS nodes which drive the vision hardware are run on a dedicated processor on platform [58]. This dedicated processor was necessary because the vision software proved much too demanding on the standard Pioneer computer. Since transmitting images and navigation data are both bandwidth-intensive, we attempt to only transmit visual data to the workstation when the platform is stationary or during teleoperation since the system is not using autonomous navigation.

5.3 Localization/Navigation

All of the NRG platforms have been tested using the Bayesian navigation algorithm described in Section 4.4. The Turtlebot was the earliest testbed for navigation using a navigation package provided by Clearpath. Testing was also performed for vision-free navigation using the wheel odometry. Results indicated that, for a dry, clean cement floor, wheel slip is low enough to navigate over short distances in the event that vision is lost.

The Pioneer LX platform has been equipped with Bayesian navigation using odometry and depth vision. Adept has a ROS navigation package called ROSArnl, which has been modified for additional capabilities by NRG. The navigation API allows autonomous navigation commands with goals specified either by world frame pose or by a unique location label.

Future work will include developing autonomous localization recovery routines. These would be used by the robot to determine its location after rebooting from some fault. This is also known as the “kidnapped robot” problem. Localization would be based on exploring the immediate surroundings and using depth vision and the Bayesian method. Since some rooms of the vault look similar, landmarks may be needed to complete the localization. Some preliminary work has been done towards using the 2D barcodes for this purpose.

5.4 User Interface/Operator Awareness

Operator awareness is mainly provided by RVIZ, a core ROS package used for visualization. RVIZ can subscribe to ROS topics and render certain ROS standard messages in a 3D space. It can also render physical objects, such as robots, from Universal Robot Description Format (URDF) files. Figure 5-1 shows a visualization of the NRG Vaultbot in RVIZ.

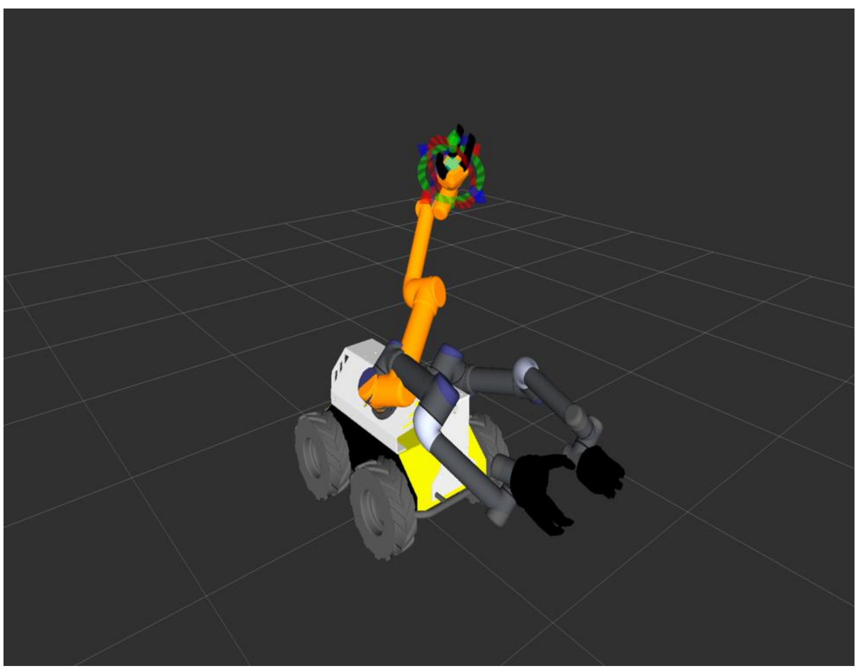


Figure 5-1: Vaultbot Rendering in RVIZ

RVIZ can interface with a plugin of the ROS package MoveIt to support motion commands and motion planning for manipulators. Figure 5-1 has an interactive marker at the end effector of one of the vaultbot’s manipulators for this purpose. RVIZ can also display data such as point clouds, depth maps, and 2D navigation maps. These features are used in the NRG demonstrations to display radiation fields, depth images of shelves and containers, and text data related to containers. This will be detailed in the following sections of this chapter.

Teleoperation is done via keyboard input using the WASD keys to send velocity commands. Camera feeds are displayed in RVIZ for feedback on position. The Pioneer can be teleoperated in a safe mode in which the ultrasonic proximity sensors will not allow the operator to drive the robot into obstacles.

Feedback on demonstration progress and robot status is mostly provided via console print statements at the workstation. Future work will involve packaging this into a GUI for greater user-friendliness.

5.5 Database Interface

The Turtlebot and Pioneer have been used to test SQL database integration using the ROS software package described in section 4.6. A preliminary demonstration was first performed using the Turtlebot. A PostgreSQL database was created on the workstation and populated with fictional data about containers in the NRG mockup vault. An early version of the `inventory_mgmt` package, called `barcode_recognition`, was added to the autonomous radiation surveying demonstration described in section 5.6.2. Containers were set up in the mock vault with unique identifying barcodes taped to the front. This resulted in a demonstration in which the Turtlebot drove to a series of waypoints in front of shelves in the mock warehouse, took radiation readings at the locations, and displayed the detected barcodes in RVIZ. The workstation was located at the other end of the lab, and the operator could watch the system's progress via the camera feed in RVIZ.

The new `inventory_mgmt` package has been installed on the Pioneer. A new demonstration has been created which incorporates the added features such as querying of locations to determine what containers are expected there. The system also identifies when containers are missing or misplaced and will modify the database accordingly. This demonstration is described in greater detail in Chapter 6.

5.6 Radiation Surveying

5.6.1 Basic Integration Test

A ROS driver was written that enables integration of the measurement hardware with the robot platform. The instrument is a basic Geiger-Mueller tube built from a student kit. It is powered by a 9V battery.

The voltage pulse output of the GM tube is monitored by an Arduino Uno microcontroller, which publishes the data as a ROS topic. The Arduino runs on a ROS library written in the Arduino IDE language [58]. Every time the Arduino detects a pulse on the GM voltage output, it produces a simple ROS message and transmits it via serial USB to an onboard processor running the client node.

The client node collects sensor data and computes the total counts, count rate, and their respective standard deviations. The system determines the presence or absence of a radiation source in the vicinity by comparing computed count rate to a preset background threshold generated for the virtual lab or archived from previous inspections using the hardware.

The basic benchtop test setup is shown in Figure 5-1. Two Cl-36 sources with a combined activity of 2.44 μCi [90.28 kBq] were placed beside the GM tube. The USB output was connected to a desktop PC running the ROS radiation analysis node. The analysis node was then started with the background threshold set to 16 cpm (based on a 10 minute background count) and a desired confidence level of 95%. The node was allowed to run until it could determine if the measured count rate exceeded the background threshold by a statistically significant amount. Count rates measured with the source present were typically 80-100 cps, which led to a quick determination that the sources were present. This was performed five times, and then five more times after the sources were removed. In every test run the system correctly identified whether the sources were present or absent.

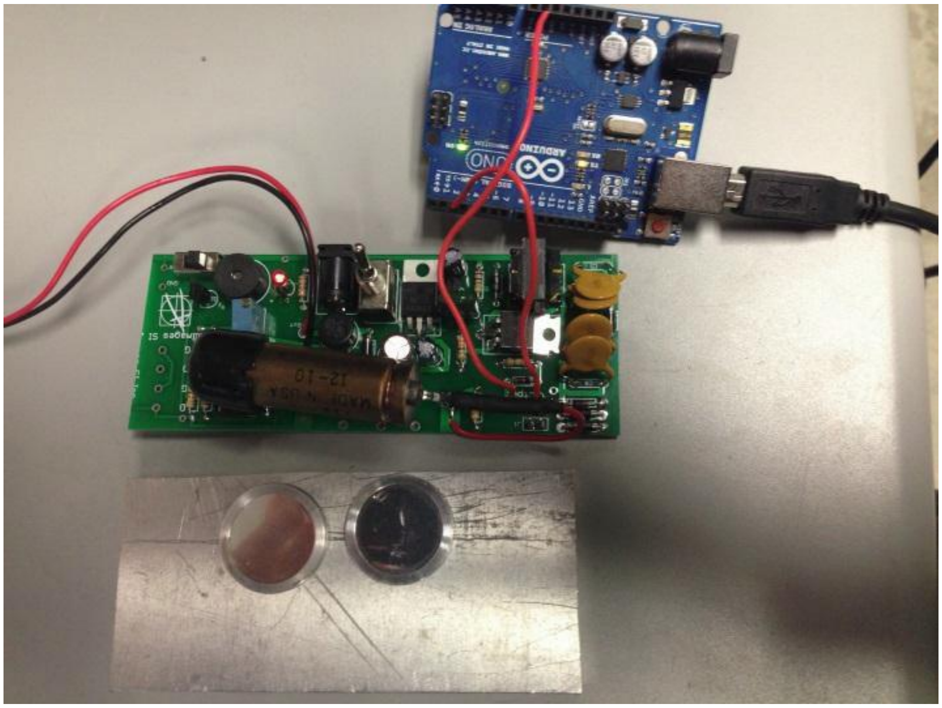


Figure 5-2: Counting Program Test

5.6.2 Turtlebot Mobile Survey Demonstration

The GM detector and Arduino were mounted on the Turtlebot 2 platform for integration testing and demonstration as shown in Figure 5-2. The detector is located at the front edge of the lower mounting surface of the robot platform.

Figure 5-3: Turtlebot Integration with GM Detector

A program was developed to perform autonomous surveying of an area using the Turtlebot. The operator first performs a teleoperated survey, and the poses of each measurement are recorded. Later, the operator can command the system to repeat the same series of measurements. The platform will create path plans to each of the survey points using the ROS `move_base` navigation package and move to each in sequence. At each point it will perform the same radiation reading as was done by the operator. The measurement will last until the system determines if it exceeds or does not exceed the background threshold by a specified confidence level. If the conclusion is different than the stored result from the teleoperated survey, the system will alert the operator via an on-screen notification. This means that in a live application, operators can set the system to perform periodic sweeps of predetermined points and be informed of any anomalous changes to the radiation profile of the area. Unexpected radiation changes can signify problems such as contamination, airborne radiation leakage, removed material, or process equipment malfunction, and so are of interest to operators. The use of robot navigation and pose storage provides high repeatability between surveys at different times, allowing the operator to make meaningful comparisons of data. RVIZ also places 3D markings on the map, with the radiation intensities indicated by marker color (Figure 5-3). This allows the operator to visually interpret the radiation profile in 3D space.

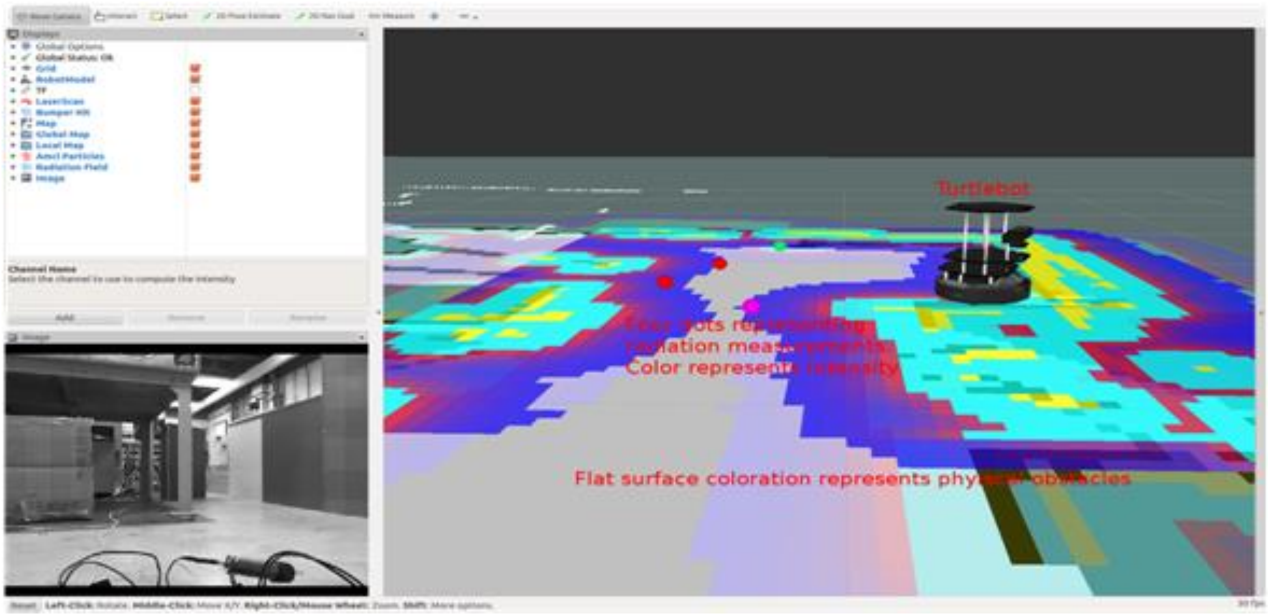


Figure 5-4: RVIZ Visualization of Radiation Measurements

5.6.2 Pioneer LX Integration

The new Pioneer LX platform has been fitted by the NRG with an alpha radiation detector [59]. It is mounted on a boom arm extending off the right side of the body (Figure 1-5). The sensing aperture is pointed directly down at the floor, at a distance of approximately 0.25 in. The sensor communicates with the main onboard computer via a serial port and the rosserial port monitoring package. The serial data is received by a ROS node which then simply publishes True or False over an '`alphadetection`' topic depending on whether or not any counts were recorded. The new demonstration described in Chapter 6 makes use of this sensor to demonstrate interrupt behavior upon detection of floor contamination.

Chapter 6: Demonstration

A mobile robot demonstration was performed which combines the features detailed in Chapters 4 and 5, including autonomous navigation, radiation detection, and object recognition.

6.1 Demonstration Goals

The demonstration simulates tasks that would be performed in the real vault. The tasks are designed to encompass each of the software development areas:

- Radioactive contamination detection.
- Barcode detection for object identification and robot localization.
- External inventory database operations.
- Spatial mapping and autonomous navigation using vision.
- Use of GOAP to manage a multi-step task and react to interrupt conditions.

The demonstration should emphasize the benefits of automation outlined in Chapter 1, including labor reduction, repeatability, and accuracy. Container recognition using barcode analysis shows the system's usefulness with respect to inventory tasks, and the ability to instantly detect when a can is missing or misplaced fulfills a valuable security role. Autonomous database management will prevent and correct wrong entries caused by human error.

6.2 Task

The demonstration task mimics an inventory inspection operation performed in the vault. The Pioneer LX platform will be used. The platform will autonomously navigate to a set of shelves in a mockup warehouse. At each shelf, it will detect and identify any containers present. Identification will be based on barcode detection. Database queries will retrieve the information related to the containers, and a location query will pull a list of expected containers at each shelf. Discrepancies between the expected container locations and found locations will be flagged for the operator, and the database will be modified to reflect the actual vault situation. The robot will begin and end the demonstration in the Adept docking station.

If unexpected radiation is detected, the system will alert the operator and pause work until the alarm is cleared by the operator. An audible alarm will play.

At some point during the survey, a fake signal will be sent to the robot to make it believe its battery is low. This will test its interrupt and re-planning capability. The platform should go to the docking station and charge. At that point a high battery signal will be sent to make it stop charging and resume the survey.

An obstacle will also be placed in the robot's path to prevent it from reaching one or more of the survey points. The system should recognize that the goal is unreachable and continue to the next survey point. At the end of the survey it will re-attempt any missed survey points.

6.2.1 GOAP Planning Model

The task plan will be generated using the GOAP method in the `task_planning` package. The only user instruction is to provide the initial worldstate and the goal worldstate. The planner is configured as shown below:

Atoms:

- `is_docked` – Is the robot in the docking station?
- `battery_high` – Is the battery charge above the high threshold (90% for this demo)?
- `battery_low` – Is the battery charge below the low threshold (10% for this demo)?
- `is_at_cabinet_X` – Is the robot in front of cabinet X?
- `cabinet_X_surveyed` – Has cabinet X been surveyed?
- `alpha_detected` – Has a new alpha contamination source been detected?

Actions:

- `dock`
    - Preconditions: `is_docked = false`
    - Postconditions: `is_docked = true`

- `charge_battery`
    - **Preconditions:** `battery_high = false; is_docked = true`
    - **Postconditions:** `battery_low = false; battery_high = true`
- `drive_to_cabinet_X`
    - **Preconditions:** `battery_low = false; alpha_detected = false`
    - **Postconditions:** `is_at_cabinet_X = true` for this cabinet, and false for all other cabinets
- `survey_cabinet_X`
    - **Preconditions:** `is_at_cabinet_X = true; cabinet_X_surveyed = false`
    - **Postconditions:** `cabinet_X_surveyed = true`
- `handle_alpha`
    - **Preconditions:** `alpha_detected = true`
    - **Postconditions:** `alpha_detected = false`

The 'X' in some of the atoms and actions indicate that there is one for each of the cabinets/shelves in the demonstration. The number of cabinets is given to the planner setup function so it can generate the appropriate sequences of these actions and atoms.

The initial and goal worldstates are shown below:

	is_docked	**battery_high**	**battery_low**	**is_at_cabinet_X (all)**	**cabinet_X_surveyed (all)**	**alpha_detected**
Initial	T	X	X	F	F	F
Goal	T	X	X	X	T	F

T – True; F – False; X – Don't Care/Not specified

Table 6-1: Initial and Goal Worldstates

At the start of the demo, the system will print the action plan so the operator can verify that it is being followed. It will also display new plans that are created in response to interrupt conditions (detailed below).

6.2.1.1 Interrupt Conditions

The planner will respond to the following interrupt conditions. These events will modify a worldstate atom and call the planner function to create a new plan which accounts for the condition.

Floor contamination – The system has detected alpha counts exceeding a specified alarm level. The robot will stop, play an audio alarm, drive forward until it no longer is detecting alpha counts, and continue the survey. This is implemented by setting atom `alpha_detected` equal to true and calling the planner.

Low battery – The battery charge falls below the specified low threshold (10% for this demonstration). The robot will go to the docking station and wait until the battery charge meets the high threshold (90% for this demo). Then it will continue the survey. This is implemented by setting atom `battery_low` equal to true and calling the planner.

Blocked drive path – The robot is unable to perform the `drive_to_cabinet` action for one or more of the cabinets. When this occurs, it sets a flag telling the program to try again after continuing on through all the remaining cabinets. The planner will be called again and the robot will attempt to survey the missed cabinets. The program will only re-attempt this once.

6.3 Performance Metrics

The performance criteria of the task are:

1. The robot travels to the correct inspection locations in the order shown on the action plan. If a survey point fails due to obstructions, it will move on to the next point.
2. The robot verifies the container's identity using barcode recognition. No false positives, false negatives, or erroneous idenfications are acceptable.
3. The system recognizes when container locations are inconsistent with the database. When this occurs it informs the operator and modifies the database to reflect reality.
4. Database fields for the containers are retrieved and displayed to the operator. The displayed information must be correct.
5. When floor contamination is detected, the system stops, plays an audio alarm, and continues the survey after being authorized by the operator.
6. The robot returns to the docking station to charge the battery when needed. It resumes the survey when charging is complete.
7. At the end of the first pass through the vault, the robot will re-attempt any surveys that failed the first time.
8. The robot returns to the docking station when the survey is complete.

6.4 Environment/Setup

The test environment will use the NRG mockup vault (Figures 6-1 and 6-2). The mockup vault consists of three parallel rows of shelves branching off of a common corridor. Containers have been placed on a set of the shelves on the lowest level. 2D barcodes have been affixed to the sides of the container facing outward. Many shelves have multiple containers. The distribution of the containers is shown in Table 6-2. In total there are twelve containers in nine shelves, with one empty shelf.

Shelf	Type	Container ID
1	paintCan	10
2	paintCan	12
	paintCan	16
3	paintCan	7
4	smallCan	4
5	smallCan	1
6	-	-
7	paintCan	2
	paintCan	5
	paintCan	17
8	smallCan	14
9	smallCan	13
10	paintCan	6

Table 6-2: Container Distribution

The SQL inventory database is set up on the mockup vault workstation. It is populated with fictional information about the containers as shown in Table 6-3. Note that containers 1, 4, 5, and 17 are shown in different locations than in Table 6-2. This will test the ability of the system to flag and correct discrepancies. The intentionally incorrect entries are highlighted.

container_id	contents	activity	date	location_id
1	Cl-36	0.025	2005-1-8	shelf4
2	U-235	0.1	1999-2-1	shelf7
4	U-238	1.0	2001-4-26	shelf5
5	Pu-239	0.2	1980-12-4	shelf5
6	Pu-239	0.9	1987-5-10	shelf10
7	U-235	1.1	1994-9-13	shelf3
10	Pu-239	0.08	1983-11-10	shelf1
12	Cm-238	2.0	1992-6-29	shelf2
13	Am-241	0.25	1997-4-16	shelf9
14	Pu-238	1.2	1986-9-10	shelf8
16	U-238	0.67	1977-5-9	shelf2
17	U-235	0.92	1997-1-4	shelf7

Table 6-3: Fictional Container Entries in SQL Database

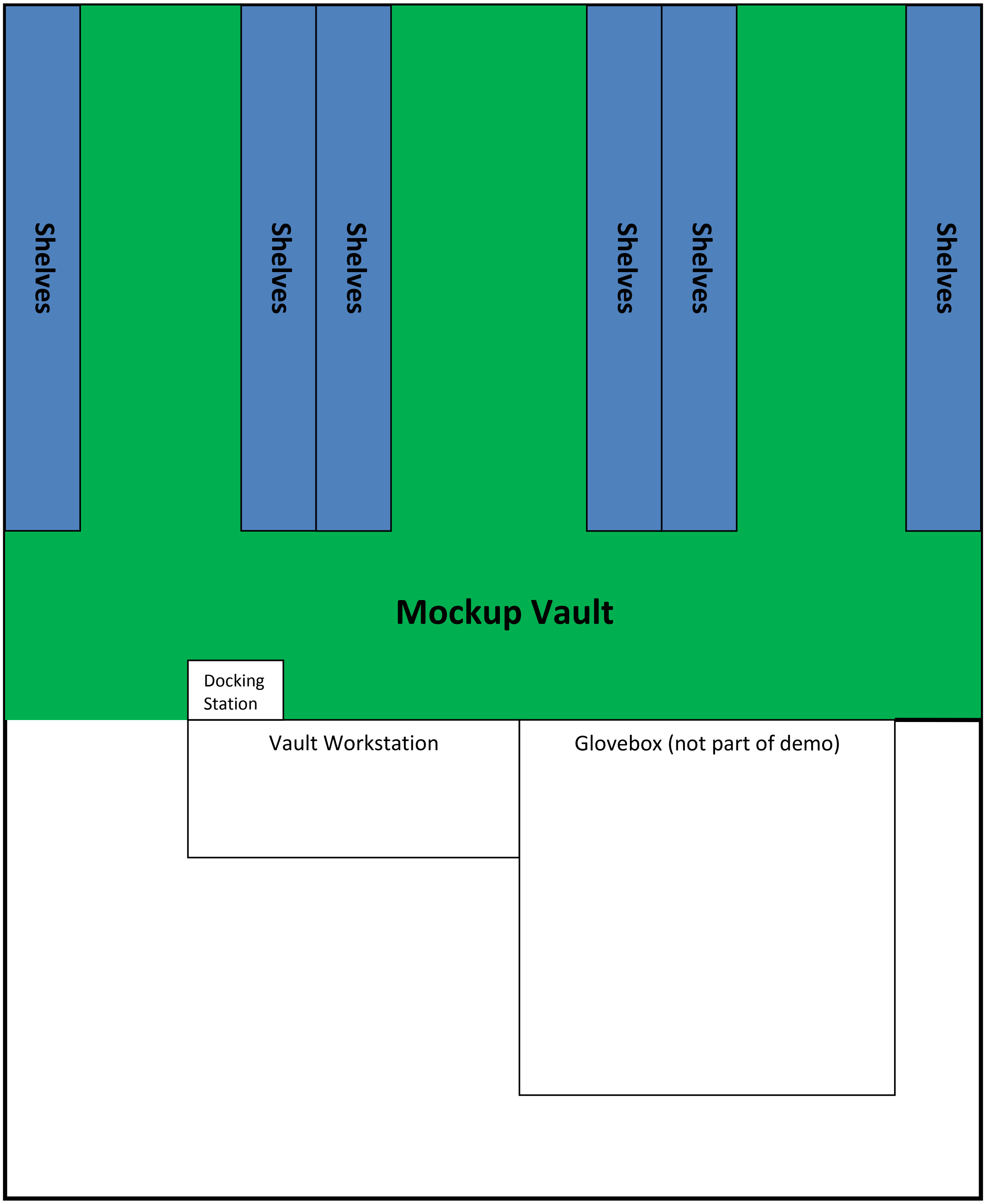


Figure 6-1: NRG Mockup Vault Diagram

Figure 6-2: NRG Mockup Vault Photographs

6.5 Demonstration Results

The demonstration was performed with mostly positive results. The resulting timeline of events is detailed below:

- Robot attempted to navigate to Shelf 1. Navigation was aborted due to poor localization.
- Robot continued to Shelf 2 and the survey was successful. Containers 12 and 16 were found there as expected.
- A fake low battery signal was sent by the operator to make the system pause the survey and return to the dock to recharge. Once it was docked and charging, the operator sent a high battery signal to make the robot resume the survey.
- The robot re-attempted Shelf 1 and was successful. Container 10 was found.
- The robot attempted to survey Shelf 3 but was unable to reach the goal pose. This is because Shelf 3 is in a corner. Interior corners make pathfinding difficult because the navigation algorithm extrapolates a collision model around the robot.

- Survey of Shelf 4 succeeded. Container 1 is expected to be at Shelf 4 but was missing. Container 4 was found, but was expected at Shelf 5. The system corrected the location of Container 4 and marked Container 1 as "MISSING".
- On the way to Shelf 5, alpha floor contamination was detected. The robot halted, played an audible alarm, and queried the operator for acknowledgment of the contamination. The survey continued after acknowledgement was given by the operator.
- After clearing the contamination alarm, the system re-attempted Shelf 3. The survey was successful and Container 7 was found.
- Survey of Shelf 5 was successful. Container 1 was found, and its location was changed from "MISSING" to Shelf 5. Container 5 was not found and was marked "MISSING."
- Navigation to Shelf 6 failed because the operator intentionally blocked the path with a board. System continued to Shelf 7.
- Survey of Shelf 7 was successful. Containers 2 and 17 were found as expected. Container 5, which was missing from Shelf 5, was found and its location was corrected.
- Survey of Shelf 8 was successful. Container 14 was found.
- Navigation to shelves 9 and 10 failed due to the corner collision issue.
- Robot returns to dock and assesses results of survey.
- Robot re-attempts the surveys of the failed shelves. The obstruction to Shelf 6 has been cleared since the initial failure, and the new attempt succeeded. No containers were found.
- Re-attempts of Shelves 9 and 10 fail again due to navigation.
- The system returned to dock and shut down.

In the end, the system successfully navigated to eight of the ten shelves and surveyed them. All containers at the surveyed shelves were detected via barcode. No containers were misidentified. In total, ten of the twelve containers were successfully inventoried. RVIZ displayed the correct contents, activity, and date of each found container per Table 6-3. RVIZ also showed correct depth images of the surveyed shelves. Table 6-4 summarizes these results.

Shelf	Actual Containers Present	Containers Present Per Database	Found Containers	Database Updates	Result
1	10	10	10	None	PASS
2	12, 16	12, 16	12, 16	None	PASS
3	7	7	7	None	PASS
4	4	1	4	1 flagged MISSING, 4 moved to Shelf 4	PASS
5	5	1	1	1 moved to Shelf 5, 5 flagged MISSING	PASS
6	None	None	None	None	PASS
7	2, 5, 17	2, 17	2, 5, 17	5 moved to Shelf 7	PASS
8	14	14	14	None	PASS
9	13	13	None (Not Surveyed)	None	FAIL
10	6	6	None (Not Surveyed)	None	FAIL

Table 6-4: Demonstration Survey Results

6.6 Demonstration Results Discussion

The results of the demonstration are encouraging. The GOAP task planning package performed perfectly. The initial survey plan covered all the necessary survey points in a reasonable order and included the return to the dock at the end. It also re-planned correctly in response to the interrupt conditions including a low battery warning, floor contamination, and an obstructed drive path. It also reacted well to unplanned faults such as localization and path planning failures. It is likely that the system could loop this demo without supervision for hours with no fatal errors.

The barcode recognition and database interface worked perfectly. The barcode analyzer produced no false positives, false negatives, or misidentifications of containers. The recognition response time upon activation of the camera feed was less than a second. The database server handled all query and update commands with no errors or noticeable response time. The system correctly flagged when a container was misplaced or missing in every case.

The only deficiency was in the navigation to certain points. The conservative collision model makes path planning difficult in confined spaces and near interior corners. In past versions of this demonstration, which used a hardcoded script for task execution, this problem was solved by simply sending the navigation goal again and again until success. The GOAP approach reacts to failed tasks dynamically, however, and so it moves on to the next survey point. This could be solved via the brute force approach of looping the survey until all survey points have succeeded. A more elegant solution might involve teaching the navigation system to retry navigation goals if it is close to success, but to move on otherwise. Tweaking the collision model to be less conservative may also help, although this has safety tradeoffs.

The entire survey took 14 minutes. This is longer that it would take a trained human to verify correct locations of the containers in this mock setup. However, the robot performs the additional work of pulling and displaying container information from the database, making database modifications as necessary, and sweeping the floor for contamination. Accounting for the automation of these tasks, the robot saved an estimated 30-45 minutes of labor. It should also be noted that the robot was deliberately slowed by various interrupt conditions in the demonstration. It would perform the task faster otherwise. Improving the navigation issues would also speed the demo by removing the need for multiple attempts at survey locations.

Chapter 7: Conclusions and Future Work

7.1 Summary

The nuclear research field faces unique challenges related to worker and public safety that result in major costs and restrictions on work. Traditional techniques for radiation protection have reached a point of quickly diminishing returns as facilities face continued regulator demands for reduced dosage. Remote robotics is an emerging technology that has the potential to provide innovative solutions which can reduce exposure to hazardous conditions. Machine automation also offers a technological solution to the issue of a limited labor pool that plagues the nuclear industry and weapons complex by reducing the need for human workers in time consuming or menial tasks. Research into artificial intelligence produces increasingly sophisticated methods of performing complex tasks with reduced or eliminated human involvement.

The University of Texas Nuclear Robotics Group has developed technologies and methods related to mobile non-contact tasks. Areas of work include teleoperation, radiation measurement and mapping, object recognition, inventory management, and task planning. A hardware demonstration was then performed to show the integration of these areas on a single platform and perform a representative vault inspection activity.

These efforts are aimed at deploying a robotic system in the nuclear materials vault in TA-55 at Los Alamos National Laboratory, with the goal of reducing worker radiation dosage.

7.2 Future Work

Although the Chapter 6 demonstration provides a good baseline of necessary technologies for initial testing/training at LANL, more sophistication and flexibility is needed before the system can be trusted in a live nuclear environment. Future work will consist of expanding the demonstration with additional system capabilities and functions, as well as packaging the software in a form which can be used by the actual vault operators. The GOAP task planner developed by NRG would benefit from the addition of Hierarchical Network-Task (HTN) planning concepts, discussed in Section 3.4. This would allow related or dependent actions to be grouped together into compound actions. This will also help with human readability of action plans generated by the system, and make it easier to implement manual changes to them.

The radiation mapping and visualization work discussed in Section 5.6.2 is also planned for inclusion in the Chapter 6 demonstration. This would involve lab radiation sources placed near inspection locations and new beta/gamma instrumentation on the platform to measure the count rates. Radiation histories, associated with those containers in the database, would be compared to current readings for consistency. Visualizations of radiation data in RVIZ will also be included. A generalized floor sweep algorithm for alpha contamination is also needed.

Development of less technically demanding interfaces is needed prior to operator testing at Los Alamos. The current command line interface requires a great deal of knowledge of ROS and programming. While this is acceptable and even beneficial during laboratory development by NRG robotics engineers, GUIs and launchers that are intuitive for vault technicians must be developed. Troubleshooting support must also be provided that does not require the user to parse arcane C++ error messages. The effectiveness of the new interfaces must then be verified by live testing with the intended operators.

The object recognition process will be improved by combining multiple verification modes (barcodes, geometry, RFID, etc.) to increase the overall reliability. Pose verification of objects is also planned. This would involve the system determining the explicit pose of the containers in the world frame and comparing to expected poses in the database. This has applicability to security inspection activities in the vault. Barcodes also have potential for use in robot localization.

Longer term, it is desired to perform contact tasks with the system. This would include depositing containers in the vault or retrieving them. Other NRG members have performed work with object grasping, and manipulation which will contribute to this effort. Challenges to this goal include arm path planning, grasp planning, and grasp verification. It is also necessary to consider criteria for acceptable placement of the base for a given manipulation. The most difficult part is likely to be latches on cabinets and doors behind which the containers are kept. The current devices are intended for human use, and require a great deal of dexterity and coordination which is not currently available with robotic end effectors. Changes to the vault setup or development of specialized robotic methods will be necessary.

The eventual goal is to deploy a robot with full manipulation capability and autonomy. The initial deployments, however, will involve non-contact activities only and utilize teleoperation. Acquisition of new platforms and hardware and development of more reliable and sophisticated automation will be an ongoing process at Los Alamos and at UT for many years.

APPENDIX A: Demonstration Code Listing

This appendix contained an abridged code listing for the Chapter 6 demo. Sections representing points of interest from the demo have been pulled from the complete log, and are shown here.

1. System initializes and attempts to survey the first point. System aborts the navigation due to mis-localization. System moves on to shelf 2.

```
WaypointDrive::runNextAction - Preconditions not met for action survey_cabinet_1. Skipping
rosarnl_node: Received goal 166mm, 917mm, 177deg
Running action survey_cabinet_1
Goal Stopped
Robot localization request
rosarnl_node: Localize init (global_localization service) request...
rosarnl_node: Running ROS node...
rosarnl_node: Creating publisher for laser S3Series_1
Motor State: 1
rosarnl_node: publishing now server mode: Stop
rosarnl_node: publishing new server status: Stopped
rosarnl_node: publishing new motors state: yes.
Sending to X:0.0622 Y:0.0337 Z:0.0000 W:0.0126 Z:0.9999
Running action drive_to_cabinet_1
Running action plan
```

2. Survey of shelf 2 succeeds.

```
Expected marker 16 found
Expected marker 12 found
Expected marker: 16
Expected marker: 12
Connected to barcode process server
Loading expected containers for shelf shelf2
Connected to inventory mgmt
Starting Visual Recognition
Classifier subscribing
Calling to process shelf 2
Running action survey_cabinet_2
Goal Reached
rosarnl_node: publishing new server status: Arrived at point
rosarnl_node: publishing now server mode: Goto point
rosarnl_node: publishing new server status: Going to point
Sending to X:0.1657 Y:0.9172 Z:0.0000 W:0.0234 Z:0.9993
Running action drive_to_cabinet_2
```

3. Low battery signal is sent to interrupt the system. System goes to docking station to charge. High battery signal is then sent and the system continues the survey. After charging, the system re-attempts the survey of shelf 1 and succeeds.

```
Expected marker 10 found
Expected marker: 10
Expected marker: 0
Loading expected containers for shelf shelf1
Connected to inventory mgmt
Classifier subscribing
Starting Visual Recognition
Calling to process shelf 1
Running action survey_cabinet_1
Goal Reached
rosarnl_node: publishing new server status: Arrived at point
rosarnl_node: publishing now server mode: Goto point
rosarnl_node: publishing new server status: Going to point
Undocked
rosarnl_node: publishing new server status: Undocking
Sending to X:0.0622 Y:0.0337 Z:0.0000 W:0.0126 Z:0.9999
Running action drive_to_cabinet_1
rosarnl_node: Received goal 62mm, 34mm, 179deg
Charging complete.
Running action charge_battery
Docking complete.
rosarnl_node: publishing new server status: Docked
Docked
rosarnl_node: publishing new server status: Driving into dock
rosarnl_node: publishing new server status: Going to dock at Dock
rosarnl_node: publishing now server mode: Dock
rosarnl_node: publishing new server status:
Robot dock request
rosarnl_node: Docking procedure request.
Running action dock
Goal Failed
rosarnl_node: publishing new server status: Failed to get to point (Failed going to goal)
Low battery level: 0.000000
```

4. The system finds that container 1 is missing from shelf 4, and it finds container 4 which the database expects to be at shelf 5. The container locations are updated in the database.

```
Container ID 4 location updated to 4
Connected to barcode process server
Modifying database info
Expected location of Container 4: shelf5
Extra containers in location 4:
Container ID 1 location updated to MISSING
```

```
Modifying database info
Containers missing from Location 4:
Missing marker 1
Extra marker 4 found
Expected marker: 1
Loading expected containers for shelf shelf4
Connected to inventory mgmt
Starting Visual Recognition
Classifier subscribing
Calling to process shelf 4
Running action survey_cabinet_4
Goal Reached
rosarnl_node: publishing new server status: Arrived at point
rosarnl_node: publishing new server status: Going to point
Sending to X:0.5297 Y:0.5889 Z:0.0000 W:0.9992 Z:-0.0409
Running action drive_to_cabinet_4
```

5. Alpha contamination is detected on the way to shelf 5. System stops, plays alarm audio and requires acknowledgement from operator to proceed.

```
Alpha contamination handled.
Alpha Gone
rosarnl_node: publishing new server status: Driving
Goal Stopped
rosarnl_node: publishing new server status: Stopped
rosarnl_node: publishing now server mode: Drive
rosarnl_node: publishing new server status: Robot lost
Goal Stopped
Press enter to acknowledge and continue survey.
Alpha contamination detected at (0.52,0.64)
rosarnl_node: publishing new server status: Stopped
rosarnl_node: publishing now server mode: Stop
Handling alpha contamination.
Running action handle_alpha
Goal Stopped
rosarnl_node: publishing new server status: Stopping
CV Stuff End
PathName: /home/pioneer/nrg-records/records/2015/12/03/1600/images/image4.png
Shelf: getFileName
Robot stop request
rosarnl_node: Stop request.
CV Stuff Start
Alpha Detected
rosarnl_node: publishing new server status: Going to point
Sending to X:2.7829 Y:0.6489 Z:0.0000 W:0.0096 Z:1.0000
Running action drive_to_cabinet_5
```

6. Container 1 is located at shelf 5 and container 5 is missing from it.

```
Container ID 1 location updated to 5
Modifying database info
Expected location of Container 1: MISSING
Extra containers in location 5:
Container ID 5 location updated to MISSING
Modifying database info
Containers missing from Location 5:
Missing marker 5
Extra marker 1 found
Expected marker: 5
Loading expected containers for shelf shelf5
Connected to inventory mgmt
Starting Visual Recognition
Classifier subscribing
Training data loaded.
cph: No models loaded from /home/plxvision/pioneer-nrg/src/nrg-ros-support/orp/data/cph.
Calling to process shelf 5
Running action survey_cabinet_5
Goal Reached
rosarnl_node: publishing new server status: Arrived at point
CV Stuff End
PathName: /home/pioneer/nrg-records/records/2015/12/03/1600/images/image3.png
Shelf: getFileName
CV Stuff Start
rosarnl_node: publishing new server status: Going to point
rosarnl_node: Received goal 2783mm, 649mm, 179deg
Sending to X:2.7829 Y:0.6489 Z:0.0000 W:0.0096 Z:1.0000
Running action drive_to_cabinet_5
```

7. The system fails to reach shelf 6 because it is blocked off.

```
Goal Failed
rosarnl_node: publishing new server status: Failed to get to point (Failed going to goal)
CV Stuff End
PathName: /home/pioneer/nrg-records/records/2015/12/03/1600/images/image5.png
Shelf: getFileName
CV Stuff Start
rosarnl_node: publishing new server status: Going to point
Sending to X:3.2186 Y:1.6593 Z:0.0000 W:0.9988 Z:-0.0498
Running action drive_to_cabinet_6
```

8. Navigation to shelves 9 and 10 fail due to the collision model. System returns to dock. It them re-attempts the shelves that failed on the first pass.

```
# Objects seen: 0
Started shelf 6 processing.
Calling to process shelf 6
Running action survey_cabinet_6
Goal Reached
rosarnl_node: publishing new server status: Arrived at point
rosarnl_node: publishing now server mode: Goto point
rosarnl_node: publishing new server status: Going to point
Undocked
rosarnl_node: publishing new server status: Undocking
Sending to X:3.2186 Y:1.6593 Z:0.0000 W:0.9988 Z:-0.0498
rosarnl_node: Received goal 3219mm, 1659mm, -6deg
Running action drive_to_cabinet_6
Docking complete.
Docked
rosarnl_node: publishing new server status: Docked
rosarnl_node: publishing new server status: Driving into dock
rosarnl_node: publishing new server status: Going to dock at Dock
rosarnl_node: publishing now server mode: Dock
rosarnl_node: publishing new server status: Stopped
Goal Stopped
Robot dock request
rosarnl_node: Docking procedure request.
Running action dock
WaypointDrive::runNextAction - Preconditions not met for action survey_cabinet_10. Skipping
Running action survey_cabinet_10
Running action drive_to_cabinet_10
WaypointDrive::runNextAction - Preconditions not met for action survey_cabinet_9. Skipping
Running action survey_cabinet_9
Goal Failed
rosarnl_node: publishing new server status: Failed to get to point (Failed going to goal)
CV Stuff End
PathName: /home/pioneer/nrg-records/records/2015/12/03/1600/images/image8.png
Shelf: getFileName
CV Stuff Start
rosarnl_node: publishing new server status: Going to point
Sending to X:6.1734 Y:1.6465 Z:0.0000 W:1.0000 Z:-0.0086
Running action drive_to_cabinet_9
```

BIBLIOGRAPHY

[1] Los Alamos National Laboratory, *Procedure P121 - Radiation Protection,* Los Alamos, 2014.

[2] Department of Energy, "Environmental Restoration and Waste Management Robotics Technology Development Program Robotics 5-Year Plan," Department of Energy, Washington, 1990.

[3] Los Alamos National Security, LLC, "National Security, Weapons Science," 2015. [Online]. Available: http://lanl.gov/science-innovation/capabilities/national-security-weapons-science/index.php. [Accessed 16 November 2015].

[4] Los Alamos National Laboratory, *Procedure TA55-DOP-091 TA-55 Nuclear Material Packaging.*

[5] Clearpath Robotics, "Turtlebot Personal Robotics Platform - Technical Specifications," 2014. [Online]. Available: http://bit.ly/1P1ZLMz. [Accessed 18 September 2015].

[6] Clearpath Robotics, "Husky Unmanned Ground Vehicle - Technical Specifications," 2014. [Online]. Available: http://bit.ly/1L2FtEH. [Accessed 14 September 2015].

[7] Universal Robots, *UR5 Technical Specifications,* Universal Robots, 05/2015.

[8] Robotiq, "Robotiq 3-Finger Adaptive Gripper Specification Sheet," 10 August 2015. [Online]. Available: http://robotiq.com/wp-content/uploads/2014/08/Robotiq-3-Finger-Adaptive-Gripper-Specifications-ES.pdf. [Accessed 8 September 2015].

[9] Adept MobileRobots, LLC, "Pioneer LX Datasheet," 2013. [Online]. Available: http://mobilerobots.com/PDFs/Pioneer%20LX%20datasheet.pdf. [Accessed 18 September 2015].

[10] Geo Systems, Inc, "ZM-4," 2015. [Online]. Available: http://zippermast.com/sam-4/. [Accessed 16 November 2015].

[11] J. J. McConnel, *Letter to Peter S. Winokur, Chairman, Defense Nuclear Facilities Safety Board,* 2013.

[12] R. Cortez et al, "Smart Radiation Sensor Management; Nuclear Search and Mapping using Mobile Robots," *IEEE Robotics & Automation Magazine,* pp. Vol 15, Issue 3, pp. 85-93, 2008.

[13] K. Nagatani et al, "Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots," *Journal of Field Robotics,* vol. 30, no. 1, pp. 44-63, 2012.

[14] Los Alamos National Laboratory, *Procedure P101-3 - Lockout/Tagout for Hazardous Energy Control,* Los Alamos, 2015.

[15] R. Cortez et al, "Experimental Implementation of Robotic Sequential Nuclear Search," *IEEE Mediterranean Conference on Control and Automation,* pp. 1-6, 2007.

[16] R. Cortez, H. Tanner and R. Lumia, "Distributed Robotic Radiation Mapping," *Springer Tracts in Advanced Robotics,* vol. 54, pp. 147-156, 2008.

[17] G. Minamoto, E. Takeuchi and S. Tadokoro, "Estimation of ground surface radiation sources from dose map measured by moving dosimeter and 3D map," in *Intelligent Robots and Systems (IROS 2014), IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, 2014.

[18] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," in *Augmented Reality*, San Francisco, 1999.

[19] T. Deyle, M. S. Reynolds and C. C. Kemp, "Finding and Navigating to Household Objects with UHF RFID Tags by Optimizing RF Signal Strength," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, 2014.

[20] B. O'Neil, "Object Recognition and Pose Estimation for Manipulation in Nuclear Materials Handling Applications," The University of Texas at Austin, Austin, 2013.

[21] A. Allevato, "Using a Depth Camera for Object Classification in Nuclear Gloveboxes," in *American Nuclear Society Student Conference, Robotics & Remote Systems Division*, College Station, TX, 2015.

[22] C. Brabec, K. Schroeder, J. Williams, B. O'Neil, J. Hashem and M. Pryor, "REDUCING THE OPERATOR'S BURDEN DURING TELEOPERATION," in *ANS EPRRSD - 13th Robotics & remote Systems for Hazardous Environments • 11th Emergency Preparedness & Response*, Knoxville, TN, 2011.

[23] D. Bruemmer, D. Few, R. Kinoshita, C. Kapoor and M. Goza, "Dynamic Autonom for Mobile Manipulation," in *Proceedings of the 1st Joint Emergency Preparedness and Response/Robotic and Remote Systems Topical Meeting*, Salt Lake City, 2006.

[24] L. Zeng and G. M. Bone, "Mobile Robot Collision Avoidance in Human Environments," *International Journal of Advanced Robotic Systems,* vol. 10, no. 41:2013, 2013.

[25] A. Zelenak, "A Compliant Control Law for Industrial, Dual-Arm Manipulators," The University of Texas at Austin, Austin, 2013.

[26] B. Argall, B. Browning and M. Veloso, "Learning to Select State Machines Using Expert Advice on an Autonomous Robot," Carnegie mellon University, Pittsburgh, 2007.

[27] H.-W. Park, "A Finite-state Machine for Accommodating Unexpected Large," University of Michigan, Ann Arbor, 2013.

[28] J. Orkin, "Three States and a Plan: The AI of FEAR," in *Game Developers Conference*, 2006.

[29] E. Long, "Enhanced NPC Behaviour using Goal Oriented Action Planning," University of Abertay Dundee, Dundee, 2007.

[30] R. E. Fikes and N. J. Nilsson, "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence,* vol. 2, no. 3-4, pp. 189-208, 1971.

[31] Y. Xu-Guang, "HTN Planning in Team Coordination," in *International Conference on Computer Science and Artificial Intelligence*, Szechuan, 2013.

[32] US National Park Service, "Library of Congress," [Online]. Available: http://memory.loc.gov/pnp/habshaer/nv/nv0200/nv0207/photos/348687pv.jpg. [Accessed 24 11 2015].

[33] D. Lovering, "Radioactive Robot: The Machines That Cleaned Up Three Mile Island," 27 March 2009. [Online]. Available: http://www.scientificamerican.com/article/three-mile-island-robots/. [Accessed 15 November 2015].

[34] D. J. Bruemmer, J. L. Marble, D. D. Dudenhoeffer, M. O. Anderson and M. D. McKay, "Intelligent robots for use in hazardous DOE environments,," in *Performance Metrics for Intelligent Systems (PERMIS)*, Gaithersburg, MD, August 2002..

[35] R. L. Minichan, R. F. Fogle and A. D. Marzolf, "H-Canyon Air Exhaust Tunnel Inspection Vehicle Development," in *ANS EPRRSD - 13th Robotics & remote Systems for Hazardous Environments, 11th Emergency Preparedness & Response*, Knoxville, 2011.

[36] PAL Robotics, "Stockbot Datasheet," 2015. [Online]. Available: http://www.pal-robotics.com/static/palrobotics/StockBot-Datasheet.pdf. [Accessed 23 November 2015].

[37] Keonn Technologies, "AdvanRobot: RFID robot for automatic inventory," [Online]. Available: http://keonn.com/images/stories/pdf/Keonn-AdvanRobot-Data_sheet.pdf. [Accessed 23 November 2015].

[38] E. Guizzo, "Three Engineers, Hundreds of Robots, One Warehouse," 2 July 2008. [Online]. Available: http://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse. [Accessed 23 November 2015].

[39] T. Simonite, "A Warehouse Worker's Best Friend—or Replacement?," MIT Technology Review, 29

April 2015. [Online]. Available: http://www.technologyreview.com/news/537121/a-warehouse-workers-best-friend-or-replacement/. [Accessed 25 November 2015].

[40] Hatteland, "AutoStore | Automated Warehouse Robots," 2015. [Online]. Available: http://autostoresystem.com/. [Accessed 25 November 2015].

[41] Clearpath Robotics, "Youtube: Ridgeback with UR5: Omnidirectional Manipulation," 6 July 2015. [Online]. Available: https://www.youtube.com/watch?v=6OS8jtwoFCw. [Accessed 30 November 2015].

[42] Open Source Robotics Foundation, "ROS.org," 2015. [Online]. Available: http://www.ros.org/. [Accessed 19 November 2015].

[43] F. A. Haight, Handbook of the Poisson Distribution, New York, NY: Wiley, 1967.

[44] G. Gilmore, Practical Gamma-Ray Spectroscopy, 2nd Edition, Chichester: Wiley, 2008.

[45] S. Neikum, "Github: ar_track_alvar," 12 October 2015. [Online]. Available: https://github.com/sniekum/ar_track_alvar. [Accessed 23 November 2015].

[46] VTT Technical Research Center of Finland, *ALVAR – A Library for Virtual and Augmented Reality.*

[47] M. Fiala, "ARTag Revision 1. A Fiducial Marker System," National Research Council Canada, 2004.

[48] ASUSTeK Computer Inc., "Xtion Pro," [Online]. Available: https://www.asus.com/3D-Sensor/Xtion_PRO/. [Accessed 26 November 2015].

[49] J. Castaneda, B. Ong, D. Westwood and E. Yao, "Passive Inspection Using the UT Austin VaultBot," The University of Texas at Austin, Austin, 2015.

[50] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, Cambridge, MA: MIT, 2005.

[51] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik,* vol. 1, p. 269–271, 1959.

[52] G. Palem, "Condition-Based Maintenance using Sensor Arrays and Telematics," *International Journal of Mobile Network Communications & Telematics,* vol. 3, no. 3, pp. 19-28, June 2013.

[53] K. Murphy, Adaptive Computation and Machine Learning : Machine Learning : A Probabilistic Perspective, London: MIT Press, 2012.

[54] A. Stolk, "Gihub: GPGOAP," 5 October 2015. [Online]. Available: https://github.com/stolk/GPGOAP. [Accessed 23 November 2015].

[55] S. J. Russel, Artificial intelligence : a modern approach, Englewood Cliffs, N.J: Prentice Hall, 1995.

[56] T. P. G. D. Group, "PostgreSQL," 2015. [Online]. Available: http://www.postgresql.org/. [Accessed 28 September 2015].

[57] Intel, "Product Brief NUC5i7RYH Intel NUC Kit," 2015. [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/nuc-kit-nuc5i7ryh-brief.pdf. [Accessed 30 November 2015].

[58] O. S. R. Foundation, "rosserial_arduino - ROS Wiki," 24 10 2011. [Online]. Available: http://wiki.ros.org/rosserial_arduino. [Accessed 15 11 2015].

[59] Ludlum Measurments, Inc, "Model 43-1 Alpha Detector," May 2015. [Online]. Available: http://www.ludlums.com/images/stories/data_sheets/M43-1.pdf. [Accessed 30 November 2015].